

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

_____ О.В.Непомнящий
подпись инициалы, фамилия
« _____ » _____ 20 18 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Моделирование работы сети произвольной топологии, работающей
по протоколу RIP, и оценка ее устойчивости средствами OmNet++

тема проекта

Руководитель	_____ подпись, дата	<u>доцент, канд.техн.наук</u> должность, ученая степень	<u>О.А.Русанова</u> инициалы, фамилия
Выпускник	_____ подпись, дата		<u>С.В.Потылицин</u> инициалы, фамилия
Нормоконтролер	_____ подпись, дата		<u>В.И.Иванов</u> инициалы, фамилия

Красноярск 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Анализ предметной области	6
1.1 Среда моделирования OMNET++	6
1.2 Протокол RIP	10
1.3 Критерии оценки устойчивости работы сети	13
2 Моделирование экспериментов	14
2.1 Описание модели	14
2.2 Постановка экспериментов	19
2.2.1 Обрыв линии	19
2.2.2 Отключение роутера	20
3 Анализ результатов экспериментов	21
3.1 Разрыв	21
3.2 Отключение	25
ЗАКЛЮЧЕНИЕ	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	29
ПРИЛОЖЕНИЕ А. Топология сети тестируемой модели	31
ПРИЛОЖЕНИЕ Б. Поочередный запуск роутеров	34
ПРИЛОЖЕНИЕ В. Графики результатов экспериментов при разрыве соединения между роутерами	36
ПРИЛОЖЕНИЕ Г. Графики результатов экспериментов при отключении роутера.....	43

ВВЕДЕНИЕ

Первые компьютерные сети появились в шестидесятых годах прошлого столетия, унаследовав много полезных свойств от более старых и распространённых телефонных сетей. Компьютерные сети сделали общедоступными неисчерпаемые объёмы информации, созданные цивилизацией за всё время своего существования, и продолжающие пополняться со всё более растущей скоростью.

Первыми появились глобальные сети, или же Wide Area Network, WAN. Они объединили территориально рассредоточенные компьютеры. При построении глобальных сетей были отработаны многие идеи, лежащие в основе современных вычислительных сетей.

В конце семидесятых годов появились большие интегральные схемы, что привело к созданию миникомпьютеров. В результате этого, даже у небольших подразделений предприятий могли появиться собственные компьютеры, работающие автономно. Вследствие роста потребностей пользователей и необходимости в автоматическом режиме обмениваться данными с другими пользователями, появились первые локальные вычислительные сети. Позже были разработаны и утверждены стандартные технологии объединения компьютеров в сеть. – Ethernet, Archnet, Token Ring, Token Bus, и другие. Стандартизация сетевых технологий превратила процесс построения локальной сети в рутинную работу.

В конце восьмидесятых годов отчётливо проявлялись отличия между глобальными и локальными сетями по протяженности и качеству линии связи, сложности методов передачи данных, скорости обмена данными и разнообразием услуг. В результате тесной интеграции локальных (LAN), глобальных (WAN) и городских (MAN) сетей произошло взаимопроникновение соответствующих технологий [1, с. 34; 37].

Для объединения компьютеров в сеть необходимо выбрать конфигурацию физических связей, то есть топологию сети. Топология

определяет требования к оборудованию, тип используемого кабеля, возможные и наиболее удобные методы управления обменом, надежность работы, возможности расширения сети.

После выбора топологии и системы адресации необходимо решить вопрос коммутации, а значит определить информационные потоки, для которых требуется прокладывать маршруты и маршрутизацию потоков.

Протоколы маршрутизации обеспечивают поиск и фиксацию маршрутов продвижения данных через составную сеть TCP/IP [1, с. 572].

Существуют способы продвижения пакетов в составных сетях, которые не требуют таблиц маршрутизации на маршрутизаторах, одним из таких способов передачи пакетов по сети является лавинная маршрутизация, это значит, что каждый маршрутизатор передаёт пакет всем своим соседям, исключая того, от которого его получил.

Маршрутизация от источника - ещё один вид маршрутизации, не требующий наличия таблиц маршрутизации.

Большинство протоколов маршрутизации создают таблицы маршрутизации.

Различают протоколы, выполняющие статическую и адаптивную (динамическую) маршрутизацию.

При статической маршрутизации все записи в таблице имеют неизменяемый статический статус. Записи о маршрутах составляются и вводятся в память каждого маршрутизатора администратором сети вручную. При адаптивной маршрутизации, все изменения конфигурации сети отражаются в таблицах маршрутизации автоматически, благодаря протоколам маршрутизации.

Применяемые сегодня в IP-сетях протоколы маршрутизации, относятся к адаптивным, распределённым протоколам, которые делятся на две группы [1, с. 573]: дистанционно-векторные (Distance Vector Algorithm, DVA) и алгоритмы состояния связей (Link State Algorithm, LSA).

В дистанционно-векторных алгоритмах каждый маршрутизатор периодически и широковещательно рассылает по сети вектор, компонентами которого являются расстояния от данного маршрутизатора до всех известных ему сетей.

Наиболее распространённым протоколом, основанным на дистанционно-векторном алгоритме, является протокол RIP.

Алгоритмы состояния связей обеспечивают каждый маршрутизатор информацией, достаточной для построения точного графа связей сети. Все маршрутизаторы работают на основании одного и того же графа, что делает процесс маршрутизации более устойчивым к изменениям конфигурации.

Протоколами, основанными на алгоритме состояния связей, являются протокол IS-IS стека OSI (этот протокол используется также в стеке TCP/IP) и протокол OSPF стека TCP/IP.

Далее рассматриваться будет только протокол RIP.

Целью данной работы является оценка реакции сети, работающей по протоколу RIP, на изменения и набор программных модулей, описывающих работу сети.

Для достижения этой цели были поставлены следующие задачи:

- рассмотреть протокол маршрутизации сетей, в частности – протокол RIP; выявить особенности протокола RIP;
- выбрать критерии устойчивости работы сети;
- смоделировать сеть произвольной топологии, в частности топологии: «звезда», «кольцо», в среде OmNet++;
- произвести оценку критериев устойчивости сети в зависимости от ее размера [параметров, загруженности] поставив серию экспериментов;
- провести анализ полученных результатов.

На основании полученных выводов определить критичные точки в работе протокола RIP.

1 Анализ предметной области

1.1 Среда моделирования OMNeT++

В настоящее время существует широкий спектр различных программных средств моделирования и исследования сетей и систем связи: OPNET Modeler; OMNeT++; Ns-2; Ns-3; COMNET III; BONes Designer; QualNet. В перечисленных системах имеются возможности моделирования сетей, но в разной степени осуществлена их поддержка.

Для реализации предложенного способа, моделирования и проверки всех утверждений и предположений используется фреймворк OMNeT++ версии 5.2 [2].

OMNeT++ (Objective Modular Network Testbed на C++) [2] - это модульная, основанная на компонентах библиотека моделирования и C++, в основном для создания сетевых симуляторов. Разработчик - Андраш Варга (Andras Varga). Являясь продуктом с открытым исходным кодом, доступен для бесплатного использования в академических и некоммерческих целях. «Сеть» в OMNeT++ понимается в более широком смысле. Она включает в себя проводные и беспроводные сети связи; сети массового обслуживания; чипы, построенные по архитектуре сеть-на-чипе, то есть, каждое вычислительное ядро связано непосредственно только с ближайшими ядрами. С помощью модельных фреймворков, разработанных как независимые проекты, обеспечиваются специфичные функции, такие как: поддержка сенсорных сетей, беспроводных ad-hoc-сетей или беспроводных динамических (самоорганизующихся) сетей, интернет-протоколов, моделирования производительности, фотонных сетей и др. OMNeT++ предлагает интегрированную среду Eclipse, графическую среду выполнения и множество других инструментов. Существуют расширения для моделирования в реальном времени, эмуляции сети, интеграции с базами данных, интеграции с SystemC и ряда других функций.

Хотя OMNeT ++ не является сетевым симулятором, он получил широкую популярность в качестве платформы сетевого моделирования в научном сообществе, а также в промышленных условиях и создании большого сообщества пользователей.

OMNeT ++ предоставляет архитектуру компонентов для моделирования сети. Компоненты (модули) запрограммированы на C ++, а затем собраны в более крупные компоненты и модели с использованием языка высокого уровня (NED). OMNeT ++ имеет обширную поддержку графического интерфейса, и благодаря своей модульной архитектуре ядро моделирования (и модели) может быть легко внедрено в пользовательские приложения [2].

OMNeT ++ работает под управлением Windows, Linux, Mac OS X и других Unix-подобных систем и состоит из следующих компонентов:

- библиотека ядра моделирования;
- язык описания топологии NED;
- IDE на основе платформы Eclipse;
- GUI для выполнения моделирования Tkenv, собирается в исполняемый файл симуляции;
- пользовательский интерфейс командной строки для выполнения моделирования (Cmdenv);
- утилиты (инструмент создания makefile и т. д.);
- документация, моделирование образцов и т. д.

OMNeT++ предоставляет эффективные инструменты для пользователя, чтобы описать структуру реальной системы. Некоторые из главных особенностей:

- иерархически вложенные модули;
- модули являются экземплярами типов модулей;
- модули связываются сообщениями по каналам;
- гибкие параметры модуля.

Модель OMNeT++ состоит из следующих частей [2]:

- описания топологий на языке NED (файлы с расширением .ned), которые описывают структуру модуля с параметрами, шлюзами и т. д. Такие файлы могут быть созданы с помощью любого текстового редактора, но IDE OMNeT++ обеспечивает поддержку двустороннего графического и текстового редактирования;

- определения сообщений (.msg файлы). Можно определить различные типы сообщений и добавить поля данных в них. OMNeT++ автоматически переведет определения сообщений в полноценные классы C++;

- исходные коды модуля представляют собой C++ файлы с расширением .h или .cc.

Система моделирования предоставляет следующие компоненты:

- ядро моделирования, которое управляет моделированием и классовыми библиотеками моделирования, написано на C++, скомпилировано в общие или статические библиотеки;

- пользовательские интерфейсы, используются при выполнении моделирования, чтобы облегчить отладку, демонстрацию или пакетное выполнение симуляций, написаны на C++, скомпилированы в библиотеки.

Программы моделирования построены из вышеуказанных компонентов. Сначала .msg файлы преобразуются в код на C++ с использованием компилятора сообщений `orpc_msgc`. После все исходники на C++ компилируются и собираются с ядром моделирования и библиотекой пользовательского интерфейса для формирования исполняемого файла моделирования или общей библиотеки. NED-файлы загружаются динамически в исходных текстовых формах при запуске программы моделирования. Программа моделирования может быть скомпилирована как отдельный исполняемый файл так, что она может быть запущена на других компьютерах, где OMNeT++ не установлена. Или симуляция может быть создана как общая библиотека, тогда общие библиотеки OMNeT++ должны присутствовать в системе [2].

После запуска программа считывает все файлы NED, потом – конфигурационный файл (обычно называется `omnetpp.ini`). Этот файл содержит параметры, которые определяют, как выполняется моделирование, значения параметров модели и т. д. В файле конфигурации также можно задать несколько выполнений моделирования; в простейшем случае они будут выполнены программой моделирования один за другим.

Процесс построения и запуска программ моделирования представлен на рисунке 1.

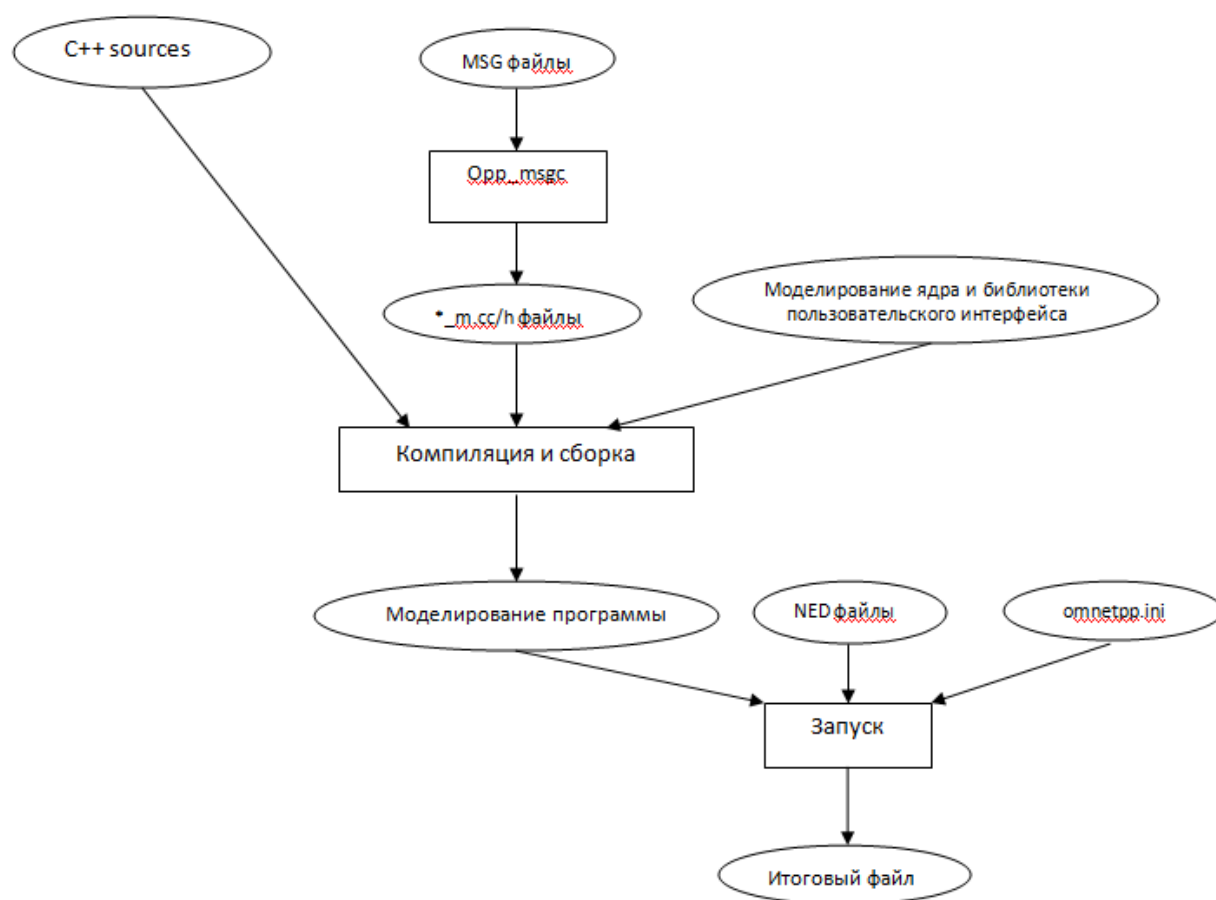


Рисунок 1 – Обзор процесса построения и запуска программ моделирования, созданных в OMNeT++

Выходные данные моделирования записываются в файлы результатов: выходные файлы векторов (запись данных во время выполнения моделирования), выходные файлы скаляров (итоговые результаты, вычисленные во время моделирования и записанные после завершения моделирования), выходные файлы, определенные пользователем. OMNeT++

содержит интегрированную среду разработки, которая обеспечивает богатые возможности для анализа этих файлов. Выходные файлы – это линейно-ориентированные текстовые файлы, что дает возможность обрабатывать их с помощью различных инструментов и языков программирования (в т. ч. MATLAB, GNU R, Perl, Python и программы для работы с электронными таблицами).

OMNeT++ может быть расширен с помощью специальных библиотек моделирования, наиболее распространенной является INET Framework, исходный код которой открыт.

INET Framework [3] содержит протоколы IPv4, IPv6, TCP, RIP, SCTP, UDP и несколько моделей приложений. Также в структуру входит модель MPLS с сигнализацией RSVP-TE и LDP. Модели на уровне канала - это PPP, Ethernet и 802.11. Статическая маршрутизация может быть настроена с использованием сетевых автоконфигураторов, или же можно использовать реализации протокола маршрутизации. INET Framework поддерживает также беспроводное и мобильное моделирование.

1.2 Протокол RIP

Протокол маршрутной информации (Routing Information Protocol, RIP) — внутренний протокол маршрутизации дистанционно-векторного типа. Будучи одним из наиболее ранних протоколов обмена маршрутной информацией, он до сих пор чрезвычайно распространен, подавляюще в малых вычислительных сетях, ввиду простоты реализации [1, с. 575].

Каждый маршрутизатор имеет информацию обо всех IP-сетях. Каждой сети в соответствие ставится метрика, являющаяся параметром, оценивающим расстояние до какой-либо сети. Каждая сеть, назначенная в среде RIP должна иметь IP-адрес, маску и метрику. Каждый маршрутизатор в сети имеет таблицу маршрутизации, в которой есть отдельная запись для каждой сети [4, с. 9].

Маршрутизаторы RIP довольно быстро приспосабливаются к новым маршрутам, так как об изменениях в сети они узнают от своих соседей. Уже сложнее обстоят дела с потерей маршрута, ведь в формате сообщений протокола RIP нет поля, указывающего на то, что путь к данной сети больше не существует. Поэтому, для уведомления о недействительности какого-либо маршрута, используются два механизма:

- истечение жизни маршрута;
- указание специального расстояния до сети, ставшей недоступной.

Первый механизм основан на том, что каждая запись таблицы маршрутизации, полученная по протоколу RIP, имеет время жизни. Каждый раз, при поступлении сообщения RIP, счётчик жизни устанавливается в исходное состояние. Если сообщение не получено и таймер доходит до нуля, то маршрут помечается как недействительный.

Второй механизм работает, когда сообщение послать можно. Если сообщение, в котором расстояние до некоторой сети равно 16 хопам, где хоп – это участок сети между двумя узлами, то это расстояние считается бесконечным. Тогда маршрутизатор проверяет, получено ли сообщение от маршрутизатора, сообщение которого ранее послужило основанием для записи в таблице. При подтверждении этого, маршрут считается недействительным.

В настоящее время протокол RIP для IP представлен двумя версиями. RIP v.1 не поддерживает маски, т. е. он распространяет между маршрутизаторами информацию только о номерах сетей и расстояниях до них, но не о масках этих сетей, считая, что все адреса принадлежат к стандартным классам А, В или С.

Классы IP-адресов называются буквами латинского алфавита от А до Е. Классы от А до Е дают возможность задать идентификаторы для 128 сетей с 16 миллионами сетевых интерфейсов в каждой, 16384 сети с 64 тысячами устройств и 2 миллионов сетей с 256 интерфейсами. IP-адреса класса А характеризуются нулевым старшим битом адреса и восьмибитным размером

принадлежности к сети. Так как первый бит адреса равен 0, то все IP-адреса класса А будут находиться в диапазоне старшего октета от 0 до 127, который, к тому же, будет являться и номером сети. При этом нулевой адрес и 127 зарезервированы под служебные адреса, поэтому использование их невозможно. По этой причине точное количество сетей класса А равняется 126. Основной отличительной особенностью IP-адреса класса В будет значение двух старших битов, равное 10. При этом размер сетевой части будет равняться 16 битам. IP-адреса класса В начинаются в диапазоне от 128 до 191. Это является отличительной особенностью, по которой можно определить принадлежность сети к этому классу. Два байта, отведенные под адреса этих сетей, за вычетом нулевых и состоящих из единиц адресов, могут составить количество узлов, равное 65 534. Любой IP-адрес класса С начинается в диапазоне от 192 до 223, при этом номер сети занимает три старших октета. Под адреса узлов в IP-адресе сетей класса С отводится 1 байт, это всего 254 хоста.

RIP v.2 передает данные о масках сетей, поэтому он в большей степени соответствует современным требованиям.

В качестве расстояния до сети стандарты протокола RIP разрешают использовать различные виды метрик: число транзитных узлов, а также любые их комбинации. Метрика должна обладать свойством аддитивности — метрика составного пути должна быть равна сумме метрик составляющих этого пути. В большинстве реализаций RIP применяется простейшая метрика — количество транзитных узлов, т. е. промежуточных маршрутизаторов, которые пакету нужно преодолеть для достижения сети назначения [4].

На рисунке 2 показано как поступает протокол RIP при выборе маршрута: вместо трех более скоростных участков магистрали он выбирает хоть и на много хуже по скорости, но лучший по количеству переходов маршрут.

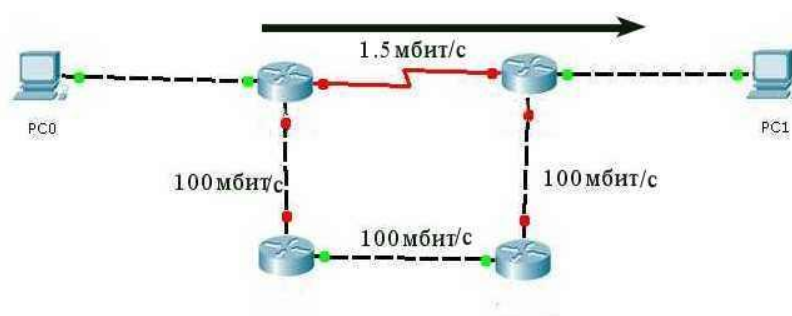


Рисунок 2 - Протокол RIP (выбор маршрута)

Учитывая особенности протокола RIP, моделирование сети произвольной топологии, работающей по данному протоколу, позволит определить критичные параметры ее эксплуатации.

1.3 Критерии оценки устойчивости работы сети

С 1 января 2011 года введена серия государственных стандартов ГОСТ 53724...Р53733-2009, формулирующая понятие качества работы сети электросвязи как совокупность показателей качества производимых услуг на различных участках и по сети в целом в соответствии с техническими требованиями к оборудованию и каналам связи [5; 6].

В соответствии с ГОСТ 53724-2009 [5]:

Устойчивость сети - это способность сохранять целостность в условиях эксплуатации.

Целостность - способность взаимодействующих сетей связи обеспечить установление соединения и передачу информации.

Приказом № 113 от 23 сентября 2007 года "Об утверждении требований к организационно-техническому обеспечению устойчивого функционирования сети связи общего пользования" [7] определены

технические нормы на показатели функционирования сетей передачи данных, такие как:

- средняя задержка передачи пакетов информации;
- отклонение от среднего значения задержки передачи пакетов информации;
- коэффициент потери пакетов информации;
- коэффициент ошибок в пакетах информации.

2 Моделирование экспериментов

2.1 Описание модели

Для проведения моделирования в OMNET++ IDE была создана сеть, модель которой представлена на рисунке 3.

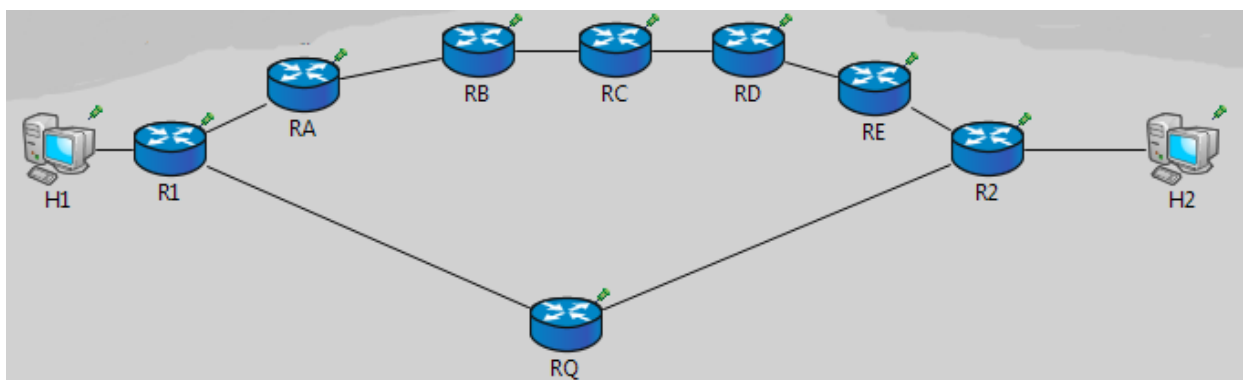


Рисунок 3 – Модель сети, написанная в OMNET++ IDE

Эта модель описана в файле `DynamicTest.ned` на языке NED, код которого приведен в приложении А.

Для создания и функционирования модели сети потребовалось в папке проекта создать соответствующие файлы:

- `DynamicTest.ned`;
- `omnetpp.ini`;
- `RIPConfig.xml`;
- `scenario1.xml`.

За создание всех компонентов сети отвечают файлы с расширением .ned. На рисунке 3 видно, что в созданную сеть входят 8 маршрутизаторов.

Создавая маршрутизатор, ему присваивается готовый класс RIPRouter, который уже включен в OMNeT++, а именно в модуль INET, где прописана вся структура роутера, принадлежащего к данному классу. Затем, в секции parameters указывается значение для @display, отвечающее за точку размещения роутера на картинке сети. После секции parameters следует секция gates, где обозначается количество интерфейсов маршрутизатора.

Пример создания роутера RB:

```
}  
    RB: RIPRouter {  
        parameters:  
            @display("p=280,69");  
        gates:  
            ethg[2];  
    }
```

При этом у каждого маршрутизатора отличается значение для @display и количество задействованных интерфейсов.

В DynamicTest.ned в секции connections создаем каналы передачи данных между маршрутизаторами, как представлено ниже:

```
connections:  
H1.ethg[0] <--> C <--> R1.ethg[0];  
R1.ethg[1] <--> C <--> RA.ethg[0];  
R2.ethg[0] <--> C <--> H2.ethg[0];  
RA.ethg[1] <--> C <--> RB.ethg[0];  
RB.ethg[1] <--> C <--> RC.ethg[0];  
RC.ethg[1] <--> C <--> RD.ethg[0];  
R1.ethg[2] <--> C <--> RQ.ethg[0];  
RQ.ethg[1] <--> C <--> R2.ethg[2];  
RD.ethg[1] <--> C <--> RE.ethg[0];  
RE.ethg[1] <--> C <--> R2.ethg[1];  
}
```

Где, C - созданный класс для канала связи. Создавая канал C через extends, указывается класс наследования ThruputMeteringChannel, у которого будут унаследованы необходимые переменные, и присвоены выбранные значения. Переменная delay отвечает за задержку, datarate отражает максимальную пропускную способность канала связи, а thruputDisplayFormat отвечает за

отображение названия канала в процессе симуляции. Так, например, создается канал C:

```
channel C extends ThruputMeteringChannel
{
    delay = 0.1us;
    datarate = 100Mbps;
    thruputDisplayFormat = "#N";
}
```

Для использования готовых классов, их необходимо подключить в файле:

```
import inet.common.lifecycle.LifecycleController;
import inet.common.misc.ThruputMeteringChannel;
import inet.common.scenario.ScenarioManager;
import inet.networklayer.configurator.ipv4.IPv4NetworkConfigurator;
import inet.networklayer.ipv4.RoutingTableRecorder;
import inet.node.inet.StandardHost;
import inet.node.rip.RIPRouter;
```

Затем, в файле `omnetpp.ini` указываем путь для плагинов, необходимых для симуляции. Время симуляции - 1000 сек. Такой временной промежуток был выбран в виду того, что на нем можно увидеть цикл работы протокола RIP.

```
#
# RIP test network.
#

[General]
description = "Dynamic test"
network = DynamicTest
#record-eventlog = true
# debug-on-errors = true
tkenv-plugin-path = ../../../../etc/plugins
sim-time-limit = 1000s
```

Также был создан файл `RIPConfig.xml`:

```
<?xml version="1.0"?>
<RIPConfig>
<interface among="R1 RQ" metric="10"/>
<interface among="R? R?" metric="1"/>
</RIPConfig>
```

В этом файле произведена настройка метрики на каждом маршрутизаторе.

Так, на роутер RQ поставлена метрика 10, что больше, чем сумма метрик между роутерами R1 – RA - ... - RE – R2.

После написания и подключения всех необходимых для моделирования файлов дизайн NED-файла выглядит в соответствии с рисунком 5.

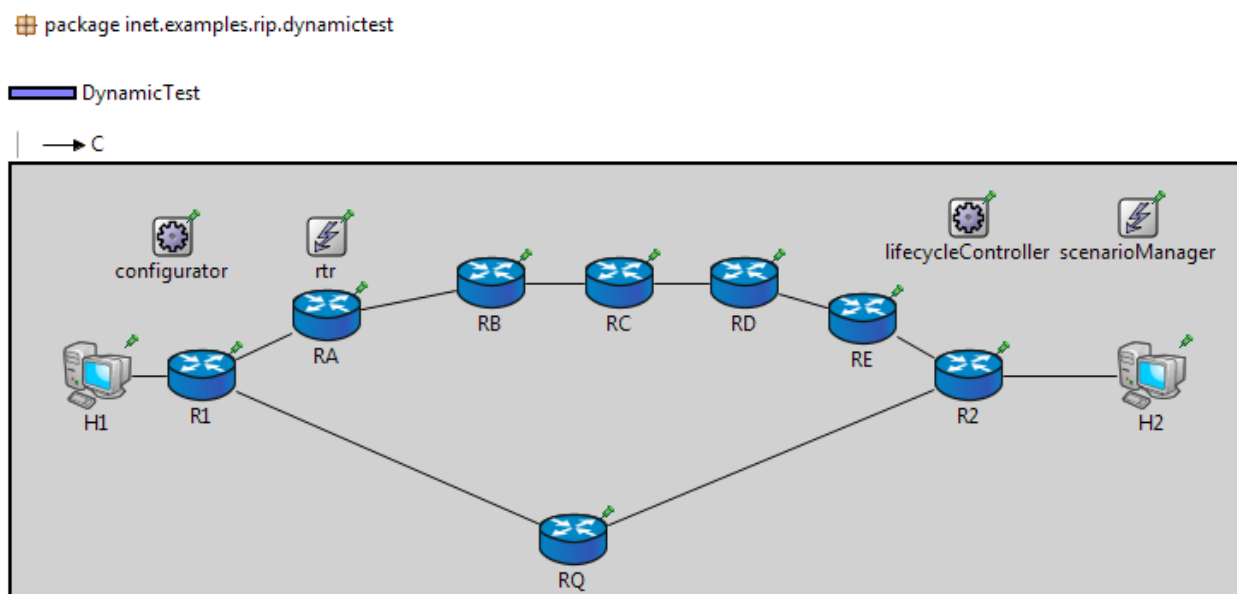


Рисунок 5 – Полученная NED-модель файла `network.ned`

На рисунке изображены все компоненты, входящие в моделируемую RIP-сеть.

При запуске моделирования проекта `riptest` происходит сборка всех входящих в него файлов и их компиляция. После чего открывается окно моделирования. Иллюстрация полученного окна моделирования представлена на рисунке 6.

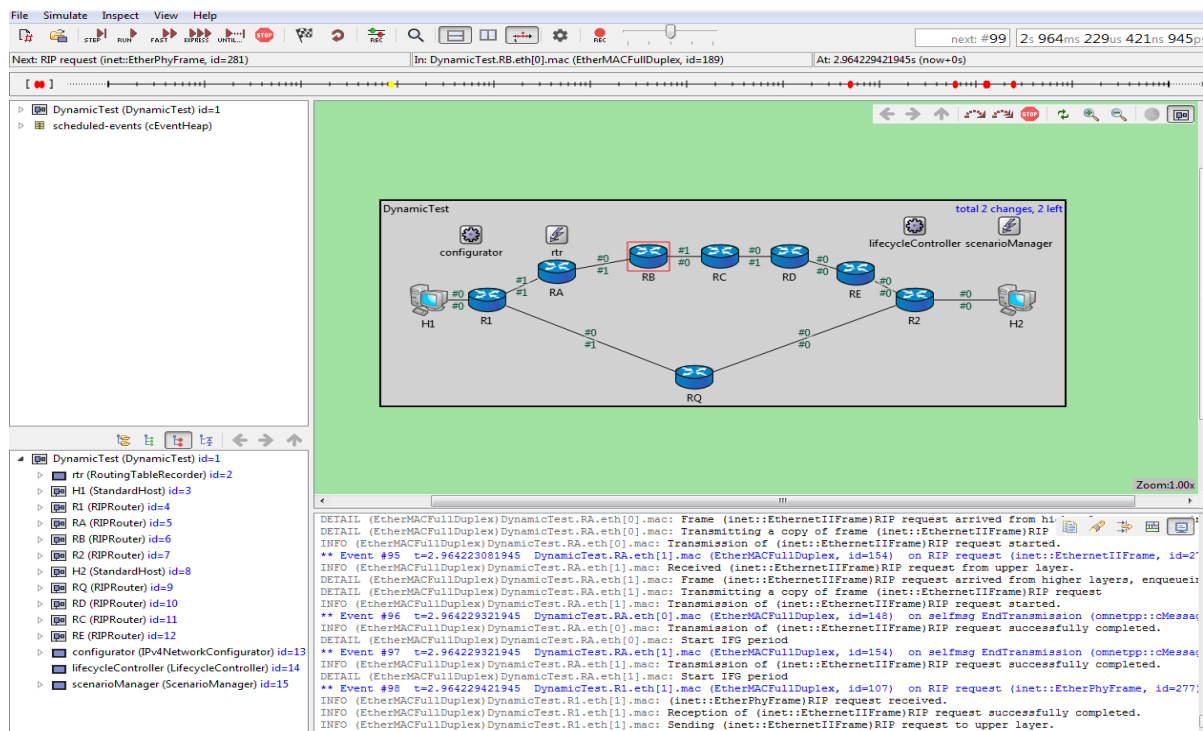


Рисунок 6 – Окно моделирования сети RIPNet

При включении самой низкой скорости симуляции протекание каждого события сопровождается анимацией в окне моделирования. Можно наблюдать инициализацию маршрутизаторов в начале симуляции, отправку пакетов данных и установление UDP-соединения с последующей передачей сегментов данных и закрытием UDP-соединения.

По окончании симуляции в папке проекта DynamicTest в файле rcvdPk:vector (packetBytes), в который были помещены все данные, собранные программой в ходе моделирования, можно получить графики работы каждого элемента сети. В частности, на графике шкала абсцисс показывает время работы сети (в секундах), шкала ординат – количество переданных пакетов данных.

Для определения исходного роутера произведен поочередный запуск маршрутизаторов, что прописано в файле scenario1.xml Приложения Б.

Начиная с 100 секунды происходит инициализация и начало передачи полезных сообщений с сетевого хаба Н1 на хаб Н2. Согласно структуре сети, RIP выберет маршрут Н1–R1–RA–RB–RC–RD–RE–R2–Н2, как маршрут с

самой малой метрикой. В какой-то момент времени происходит разрыв маршрута, сеть восстанавливает соединение, передача продолжается по маршруту H1-RQ-H2.

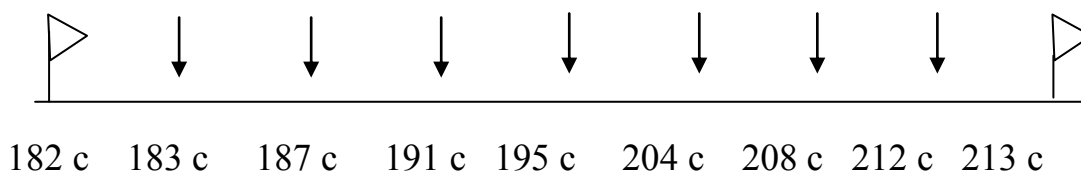
2.2 Постановка экспериментов

Цель экспериментов - определение влияния дальности разрыва на скорость восстановления связности маршрута. Разрыв – это обрыв линии или выключение роутера.

2.2.1 Обрыв линии

Был произведен обрыв линии между маршрутизаторами R1и RA, RA – RB, RB - RC, RC - RD, RD – RE, RE – R2 поочередно во временном промежутке от 182 сек. до 442 сек. Время разрыва взято таким образом, чтобы к тому моменту сеть уже стабилизировалась. Также для чистоты эксперимента была сделана поправка на время рассылки.

Ниже представлена временная диаграмма обрыва линии, t разрыва:



Для этого был создан файл `scenario1.xml`, в котором каждый раз меняли маршрутизаторы:

```
<at t="188.1">
  <disconnect src-module="RC" src-gate="ethg$o[1]" />
  <disconnect src-module="RD" src-gate="ethg$o[0]" />
</at>
<at t="388.1">
  <connect src-module="RC" src-gate="ethg[1]"
    dest-module="RD" dest-gate="ethg[0]"
    channel-
type="inet.common.misc.ThruputMeteringChannel">
```

```

        <param name="delay" value="0.1us" />
        <param name="datarate" value="100Mbps" />
        <param name="thruputDisplayFormat" value='"#N"' />
    </connect>
</at>

```

В результате проведенных экспериментов в файле rcvdPk:vector (packetBytes), в который были помещены все данные, собранные программой в ходе моделирования, были получены графики работы нескольких элементов сети.

Согласно структуре сети, изначально RIP выбирает маршрут H1–R1–RA–RB–RC–RD–R2–H2, как маршрут с самой малой метрикой. В момент времени 183 сек. происходит разрыв маршрута между маршрутизаторами R1 и RA, и начинается передача пакетов с хаба H1 на H2 по маршруту H1–RQ–H2.

По аналогии производится обрыв линии между роутерами RA – RB, RB – RC, и далее до RE – R2.

2.2.2 Отключение роутера

Для проведения экспериментов с отключением роутеров были выбраны те же временные промежутки, что и в п. 2.2.1 и создан файл scenario2.xml, в котором каждый раз меняли маршрутизаторы:

```

<at t="212">
    <tell module="lifecycleController" target="RD"
operation="NodeShutdownOperation"/>
</at>
<at t="442">
    <tell module="lifecycleController" target="RD"
operation="NodeStartOperation"/>
</at>

```

В файле rcvdPk:vector (packetBytes), в который были помещены все данные, собранные программой в ходе моделирования, были получены графики работы элементов сети.

3 Анализ результатов экспериментов

3.1 Разрыв

После проведения экспериментов по разрыву сети результаты, полученные на маршрутизаторе R1 через файл rcvdPk:vector(packetBytes) DynamicTest.RQ.udr экспортируем в Excel и строим графики, представленные в Приложении Б.

По результатам экспериментов составлены таблицы, из которых видно количество полученных байтпакетов на маршрутизаторе R1 при различных разрывах сети. Как пример, в Таблице 1 приведены результаты эксперимента при разрыве сети 183 с – 413 с.

Таблица 1 – Общая сумма байтпакетов при разрыве сети 183 с – 413 с

Данные в байтах

Время, t	разрыв R1 - RA	разрыв RA - RB	разрыв RB - RC	разрыв RC - RD	разрыв RD - RE	разрыв RE - R2
	32	32	32	32	32	32
	92	92	92	92	92	92
	32	32	32	32	32	32
	32	32	32	32	32	32
	32	32	32	32	32	32
	212	212	212	212	212	212
	92	92	92	92	92	92
	212	212	212	212	212	212
	52	52	52	52	52	52
	212	212	212	212	212	212
	52	152	132	212	212	212
	212	212	212	112	132	152
	32	72	92	212	92	72
	212	212	212	92	212	212
	212	112	112	92	72	52
	32	212	72	212	112	132
	212	52	212	212	212	212
	52	212	212	212	212	212
	212	192	212	212	212	212
	32	212	212	212	212	212

Окончание таблицы 1

	212	192	212	92	212	212
	212	192	72	92	112	132
	212	132	112	212	212	212
	212	52	212	212	72	52
	212	212	212	212	212	212

	212	212	212	212	212	212
	212	212	212	212	212	212
Итого:	12860	14868	14948	14836	14948	14948

Далее, все результаты экспериментов сведены в единую Таблицу 2.

Таблица 2 – Результаты экспериментов

Данные в байтах

Разрыв сети	Наблюдение						
	1	2	3	4	5	6	7
R1 - RA	12860	12860	13072	13000	12860	12860	12860
RA- RB	14868	14716	14888	14836	14816	14816	14868
RB- RC	14948	14948	14948	14948	14948	14876	14948
RC- RD	14836	14948	14948	14948	14856	14856	14948
RD- RE	14948	14948	14948	14948	1500	14948	14948
RE- R2	14948	14948	14816	14948	14948	14816	14816

Для того, чтобы применить критерий Крускала - Уоллиса, необходимо заменить наблюдения в s выборках их объединенными рангами. При этом первый ранг соответствует наименьшему наблюдению, а ранг n — наибольшему ($n = n_1 + n_2 + \dots + n_c$). Если некоторые значения повторяются, им присваивается среднее значение их рангов.

Для проверки гипотезы, что s независимых выборок извлечены из генеральных совокупностей, имеющих одинаковые медианы использован метод ранговых критерий Крускала – Уоллиса [8; 9]. То есть, нулевая и альтернативная гипотезы формулируются следующим образом: H_0 : $M_1 = M_2 = \dots = M_c$; H_1 : не все M_j ($j = 1, 2, \dots, c$) являются одинаковыми. Для этого необходимо знать ранги, вычисленные по всем выборкам, а s генеральных совокупностей, из которых они извлечены, должны иметь одинаковые изменчивость и вид. Для того, чтобы применить критерий Крускала-Уоллиса,

необходимо заменить наблюдения в s выборках их объединенными рангами. При этом первый ранг соответствует наименьшему наблюдению, а ранг n — наибольшему ($n = n_1 + n_2 + \dots + n_c$). Если некоторые значения повторяются, им присваивается среднее значение их рангов.

В критерии Крускала-Уоллиса средние ранги каждой из s групп сравниваются с общим рангом, вычисленным на основе всех n наблюдений. Если существует статистически значимый эффект эксперимента, средние ранги каждой группы будут значительно отличаться друг от друга и от общего ранга. При возведении этих разностей в квадрат H -статистика увеличивается. С другой стороны, если эффект эксперимента не наблюдается, статистика H теоретически должна быть равной нулю. Однако на практике вследствие случайных изменений статистика H будет ненулевой, но достаточно малой.

Критерий Крускала-Уоллиса для разностей между s медианами:

$$H = \left[\frac{12}{n(n+1)} \right] \sum_{j=1}^c \frac{T_j^2}{n_j} - 3(n+1), \quad (1)$$

где n - общее количество наблюдений в объединенных выборках, n_j - количество наблюдений в j -й выборке ($j = 1, 2, \dots, c$), T_j — сумма рангов j -й выборки.

При достаточно большом объеме выборок (больше пяти) H -статистику можно аппроксимировать χ^2 - распределением с $c - 1$ степенями свободы. Таким образом, при заданном уровне значимости α решающее правило формулируется так: гипотеза H_0 отклоняется, если $H > \chi_{\alpha}^2$, в противном случае гипотеза H_0 не отклоняется. Критические значения χ^2 - распределения вычисляются с помощью функции Excel =ХИ2.ОБР (вероятность; степени_свободы).

Нулевая гипотеза заключается в том, что нет разницы, где произошёл обрыв и удалённость обрыва не влияет на скорость восстановления.

Результаты экспериментов и ранги приведены на рисунке 7.

	Наблюдение 1		Наблюдение 2		Наблюдение 3		Наблюдение 4		Наблюдение 5		Наблюдение 6		Наблюдение 7	
	байтпакет	ранг	байтпакет	ранг	байтпакет	ранг	байтпакет	ранг	байтпакет	ранг	байтпакет	ранг	байтпакет	ранг
R1-RA	12860	3	12860	3	13072	7	13000	6	12860	3	12860	3	12860	3
RA-RE	14868	18,5	14716	8	14888	21	14836	14,5	14816	11	14816	11	14868	18,5
RB-RC	14948	31,5	14948	31,5	14948	31,5	14948	31,5	14948	31,5	14876	20	14948	31,5
RC-RD	14836	14,5	14948	31,5	14948	31,5	14948	31,5	14856	16,5	14856	16,5	14948	31,5
RD-RE	14948	31,5	14948	31,5	14948	31,5	14948	31,5	15000	42	14948	31,5	14948	31,5
RE-R2	14948	31,5	14948	31,5	14816	11	14948	31,5	14948	31,5	14816	11	14816	11
	T1=	130,5	T2=	137	T3=	133,5	T4=	147	T5=	136	T6=	93	T7=	127

Рисунок 7 – Полученные байтпакеты и ранги, при разрывах в разных частях сети и временные промежутки

В процессе преобразования 42 результатов в объединенные ранги, выясняется, что наименьшее количество байт пакетов на маршрутизатор R1 соответствует разрыву сети между R1 и RA практически во всех временных промежутках. Поскольку им соответствуют ранги от 1 до 5, то присваивается ранг 3, как среднее полученное значение. Ранг 42 присваивается самому большему результату. После присвоения рангов вычисляется их сумма в каждой группе: $T_1 = 130,5$; $T_2 = 137$; $T_3 = 133,5$; $T_4 = 146,5$; $T_5 = 135,5$; $T_6 = 93$; $T_7 = 127$.

Для проверки рангов просуммируем эти величины:

$$T_1 + T_2 + T_3 + T_4 + T_5 + T_6 + T_7 = n(n+1)/2 = 903$$

$$130,5 + 137 + 133,5 + 146,5 + 135,5 + 93 + 127 = 903$$

Используя формулу (1), вычислим H -статистику:

$$H = \left[\frac{12}{n(n+1)} \right] \sum_{j=1}^c \frac{T_j^2}{n_j} - 3(n+1) = \left\{ \frac{12}{42(42+1)} \left[\frac{130,5^2}{6} + \frac{137^2}{6} + \frac{133,5^2}{6} + \frac{146,5^2}{6} + \frac{135,5^2}{6} + \frac{93^2}{6} + \frac{127^2}{6} \right] - 3(42+1) \right\} = 1,9213732$$

Статистика H имеет приближенное χ^2 -распределение с $(c-1)$ степенями свободы. При уровне значимости α , равном 0,01, определяем величину χ_{α}^2 — верхнего критического значения χ^2 -распределения с $(c-1) = 6$ степенями свободы с использованием функции =ХИ2.ОБР($1-\alpha$; $c-1$) = 0,87209033. Как правило, уровень значимости α , берется равным 0,05. При

большом количестве производимых сравнений необходимо использовать меньший критический уровень значимости, который рассчитывается по формуле: $\alpha = 1 - 0,95^{\frac{1}{n}}$, где n – количество производимых сравнений. Для данного примера $\alpha = 1 - 0,95^{\frac{1}{42}} \approx 0,011$ [10].

Результаты вычислений приведены на рисунке 8.

Количество			c	7				
Уровень значимости			α	0,01				
Верхнее критическое значение χ^2 -распределен			χ_{α}^2	0,87209033	=ХИ2.ОБР(1 – α ; $c - 1$)			
H-статистика			H	1,9213732				
Вероятность			p(H)	0,07321409	=1-ХИ2.РАСП(G31;G28-1;ИСТИНА)			

Рисунок 8 – Результаты вычислений

Поскольку вычисленная H -статистика равна 1,9213732 и превышает критическое значение 0,87209033, нулевая гипотеза отклоняется, а значит, существует зависимость от места обрыва и удалённость обрыва влияет на скорость восстановления.

3.2 Отключение

После проведения экспериментов по отключению роутеров результаты, полученные на маршрутизаторе R1 через файл rcvdPk:vector(packetBytes) DynamicTest.RQ.udp экспортированы в Excel и построены графики, представленные в Приложении В.

По результатам экспериментов составлена Таблица 3, из которой видно количество полученных байтпакетов на маршрутизаторе R1 при отключении различных роутеров:

Таблица 3 – Количество байтпакетов при отключении роутеров

Данные в байтах

Отключенный роутер	Наблюдение						
	1	2	3	4	5	6	7
R1	11464	11464	11644	11636	11644	11676	11464
RA	13060	13008	13272	13220	13060	13060	13060
RB	14908	14776	14928	14908	14836	14908	14908
RC	14968	14968	14856	14968	14876	14968	14968
RD	14876	14968	14968	14180	14968	14856	14968
RE	14968	14968	14968	14968	14968	14836	14896
R2	14788	14920	14888	14920	14928	14908	14736

Аналогично п. 3.2 для статистического анализа оценки значимых различий использован метод ранговых критерий Крускала – Уоллиса [8; 9].

Получено 49 результатов, которым присвоены ранги.

Результаты экспериментов и ранги приведены на рисунке 9.

	Наблюдение 1		Наблюдение 2		Наблюдение 3		Наблюдение 4		Наблюдение 5		Наблюдение 6		Наблюдение 7	
	байтпакет	ранг	байтпакет	ранг	байтпакет	ранг	байтпакет	ранг	байтпакет	ранг	байтпакет	ранг	байтпакет	ранг
R1	11464	2	11464	2	11644	5,5	11636	4	11644	5,5	11676	7	11464	2
RA	13060	10,5	13008	8	13272	14	13220	13	13060	10,5	13060	10,5	13060	10,5
RB	14908	28	14776	16	14928	33,5	14908	28	14836	18,5	14908	28	14908	28
RC	14968	41,5	14968	41,5	14856	20,5	14968	41,5	14876	22,5	14968	41,5	14968	41,5
RD	14876	22,5	14968	41,5	14968	41,5	15180	49	14968	41,5	14856	20,5	14968	41,5
RE	14968	41,5	14968	41,5	14968	41,5	14968	41,5	14968	41,5	14836	18,5	14896	25
R2	14788	17	14920	31,5	14888	24	14920	31,5	14928	33,5	14908	28	14736	15
	T1=	163	T2=	182	T3=	180,5	T4=	208,5	T5=	173,5	T6=	154	T7=	163,5

Рисунок 10 – Полученные байтпакеты и ранги, при отключении роутеров в разные временные промежутки

Используя формулу (1), вычислим H -статистику:

$$H = \left[\frac{12}{n(n+1)} \right] \sum_{j=1}^c \frac{T_j^2}{n_j} - 3(n+1) = \left[\frac{12}{49(49+1)} \right] \sum_{j=1}^c \frac{1225^2}{7} - 3(49+1) = 1,3441399$$

Статистика H имеет приближенное χ^2 -распределение с $(c - 1)$ степенями свободы. При уровне значимости α , равном 0,01, определяем величину χ_{α}^2 — верхнего критического значения χ^2 -распределения с $(c - 1)$

= 6 степенями свободы с использованием функции =ХИ2.ОБР(1 – α ;с –1) = 0,872090332.

Результаты вычислений приведены на рисунке 11.

Количество			c	7			
Уровень значимости			α	0,01			
Верхнее критическое значение χ^2 -расп			χ_{α}^2	0,872090332	=ХИ2.ОБР(1 – α ;с –1)		
H-статистика			H	1,344139942			

Рисунок 11 – Результаты вычислений

Вычисленная H-статистика равна 1,3441399 превышает критическое значение 0,872090332, нулевая гипотеза отклоняется, а значит, существует зависимость от месторасположения отключенного роутера и его влияние на скорость восстановления работы сети.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы был рассмотрен протокол маршрутизации RIP. В среде моделирования OMNET++ была смоделирована сеть, работающая по данному протоколу.

С помощью библиотеки INET Framework для OMNET++ был реализован модуль, обеспечивающий работу протокола RIP.

Были проведены эксперименты по разрыву сети и отключению роутера. В результате получено подтверждение, что маршрутизаторы RIP довольно быстро приспосабливаются к новым маршрутам. Полученные данные были обработаны с применением критерий Крускала - Уоллиса. В результате чего, нулевая гипотеза о том, что нет разницы, где произошёл обрыв сети или отключение роутера, и что удалённость обрыва не влияет на скорость восстановления работы сети, было опровергнуто.

Таким образом, можно сделать вывод, что протокол RIP может быть использован в автономных сетях для быстрого перенаправления трафика по другому маршруту в случаях обрыва сети или выхода из строя роутера.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов / В. Г. Олифер, Н. А. Олифер. Изд. 4-е – Санкт-Петербург : Питер, 2012. – 944 с.
2. OmNet++ [Электронный курс] : OMNeT++ Discrete Event Simulator. - Режим доступа: <https://www.omnetpp.org/>
3. INET Framework [Электронный курс] : INET Framework . – Режим доступа: <https://inet.omnetpp.org>
4. Перевод RFC 2453, RIP. Версия 2 [Электронный курс] : Энциклопедия сетевых протоколов. – Режим доступа: http://www.protocols.ru/files/RFC/2453_RIPv2.pdf
5. ГОСТ Р 53724-2009 Качество услуг связи. Общие положения. – Введ. 01.01.2011. – Москва : Стандартинформ, 2011. – 6 с.
6. Кочеров, А. В. Функционирование единой сети связи – показатели устойчивости и текущие результаты контроля / А. В. Кочеров// 7-я международная конференция «Управление сетями электросвязи». – Москва, 2011.
7. Требования к организационно - техническому обеспечению устойчивого функционирования сети связи общего пользования : Приказ Министерства информационных технологий и связи Российской Федерации от 27 сентября 2007 года № 113. – Москва. – 2007.
8. Кобзарь, А. И. Прикладная математическая статистика / А. И. Кобзарь. - Москва : Физматлит, 2006. - 816 с.
9. Kruskal, W. H., Wallis, W. A. Use of ranks in one-criterion variance analysis / W. H. Kruskal, W. A. Wallis // Journal of the American Statistical Association. - 1953. - V.48. - P. 907-911.
10. Гржибовский, А. М. Анализ трех и более групп количественных данных / А. М. Гржибовский // Экология человека. – 2008. – № 3. – С. 50–58.

11. СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Взамен СТО 4.2-07-2012: введ. 09.01.2017. - Красноярск : ИПК СФУ, 2014. – 60 с.

ПРИЛОЖЕНИЕ А

Топология сети тестируемой модели

DynamicTest.ned

```
package inet.examples.rip.dynamictest;

import inet.common.lifecycle.LifecycleController;
import inet.common.misc.ThruputMeteringChannel;
import inet.common.scenario.ScenarioManager;
import inet.networklayer.configurator.ipv4.IPv4NetworkConfigurator;
import inet.networklayer.ipv4.RoutingTableRecorder;
import inet.node.inet.StandardHost;
import inet.node.rip.RIPRouter;

network DynamicTest
{
    parameters:
        @display("p=10,10;b=712,152");
    types:
        channel C extends ThruputMeteringChannel
        {
            delay = 0.1us;
            datarate = 100Mbps;
            thruputDisplayFormat = "#N";
        }
    submodules:
        rtr: RoutingTableRecorder {
            @display("p=184,42");
        }
        H1: StandardHost {
            parameters:
                @display("p=50,123");
            gates:
                ethg[1];
        }
        R1: RIPRouter {
            parameters:
                @display("p=111,123");
            gates:
                ethg[3];
        }
        RA: RIPRouter {
```

```

        parameters:
            @display("p=184,88");
        gates:
            ethg[2];
    }
    RB: RIPRouter {
        parameters:
            @display("p=280,69");
        gates:
            ethg[2];
    }
    R2: RIPRouter {
        parameters:
            @display("p=559,123");
        gates:
            ethg[3];
    }
    H2: StandardHost {
        parameters:
            @display("p=666,123");
        gates:
            ethg[1];
    }
    RQ: RIPRouter {
        @display("p=328,219");
        gates:
            ethg[2];
    }
    RD: RIPRouter {
        @display("p=428,69");
        gates:
            ethg[2];
    }
    RC: RIPRouter {
        @display("p=355,69");
        gates:
            ethg[2];
    }
    RE: RIPRouter {
        @display("p=497,90");
        gates:
            ethg[2];
    }
    configurator: IPv4NetworkConfigurator {
        parameters:

```



```

        @display("p=94,42");
        config = xml("<config>"
            + "<interface among='H1 R1'"
address='192.168.1.x' netmask='255.255.255.x' />"
            + "<interface among='H2 R2'"
address='192.168.2.x' netmask='255.255.255.x' />"
            + "<interface among='R*' address='10.0.0.x'"
netmask='255.255.255.x' />"
            + "<route hosts='H1' destination='*'
gateway='R1' />"
            + "<route hosts='H2' destination='*'
gateway='R2' />"
            + "<route hosts='R*' destination='224.0.0.0'"
netmask='240.0.0.0' interface='eth0' />"
            + "<route hosts='R*' destination='224.0.0.0'"
netmask='240.0.0.0' interface='eth1' />"
            + "<route hosts='R1 R2'"
destination='224.0.0.0' netmask='240.0.0.0' interface='eth2' />"
            + "</config>");
    }
    lifecycleController: LifecycleController {
        @display("p=559,31");
    }
    scenarioManager: ScenarioManager {
        @display("p=658,31");
    }
}

```

connections:

```

H1.ethg[0] <--> C <--> R1.ethg[0];
R1.ethg[1] <--> C <--> RA.ethg[0];
R2.ethg[0] <--> C <--> H2.ethg[0];
RA.ethg[1] <--> C <--> RB.ethg[0];
RB.ethg[1] <--> C <--> RC.ethg[0];
RC.ethg[1] <--> C <--> RD.ethg[0];
R1.ethg[2] <--> C <--> RQ.ethg[0];
RQ.ethg[1] <--> C <--> R2.ethg[2];
RD.ethg[1] <--> C <--> RE.ethg[0];
RE.ethg[1] <--> C <--> R2.ethg[1];
}

```

ПРИЛОЖЕНИЕ Б

Поочередный запуск роутеров

scenario1.xml

```
<at t="1">
    <tell module="lifecycleController" target="R1"
operation="NodeShutdownOperation"/>
</at>
<at t="2.1">
    <tell module="lifecycleController" target="R1"
operation="NodeStartOperation"/>
</at>

<at t="1">
    <tell module="lifecycleController" target="RA"
operation="NodeShutdownOperation"/>
</at>
<at t="5.1">
    <tell module="lifecycleController" target="RA"
operation="NodeStartOperation"/>
</at>

<at t="1">
    <tell module="lifecycleController" target="RB"
operation="NodeShutdownOperation"/>
</at>
<at t="8.1">
    <tell module="lifecycleController" target="RB"
operation="NodeStartOperation"/>
</at>

<at t="1">
    <tell module="lifecycleController" target="RC"
operation="NodeShutdownOperation"/>
</at>
<at t="11.1">
    <tell module="lifecycleController" target="RC"
operation="NodeStartOperation"/>
</at>

<at t="1">
    <tell module="lifecycleController" target="RD"
operation="NodeShutdownOperation"/>
</at>
<at t="14.1">
    <tell module="lifecycleController" target="RD"
operation="NodeStartOperation"/>
</at>
```

```

<at t="1">
    <tell module="lifecycleController" target="RE"
operation="NodeShutdownOperation"/>
</at>
    <at t="17.1">
        <tell module="lifecycleController" target="RE"
operation="NodeStartOperation"/>
    </at>

<at t="1">
    <tell module="lifecycleController" target="RQ"
operation="NodeShutdownOperation"/>
    </at>
    <at t="23.1">
        <tell module="lifecycleController" target="RQ"
operation="NodeStartOperation"/>
    </at>

    <at t="1">
        <tell module="lifecycleController" target="R2"
operation="NodeShutdownOperation"/>
    </at>
    <at t="20.1">
        <tell module="lifecycleController" target="R2"
operation="NodeStartOperation"/>
    </at>

```

ПРИЛОЖЕНИЕ В

Графики результатов экспериментов при разрыве соединения между роутерами

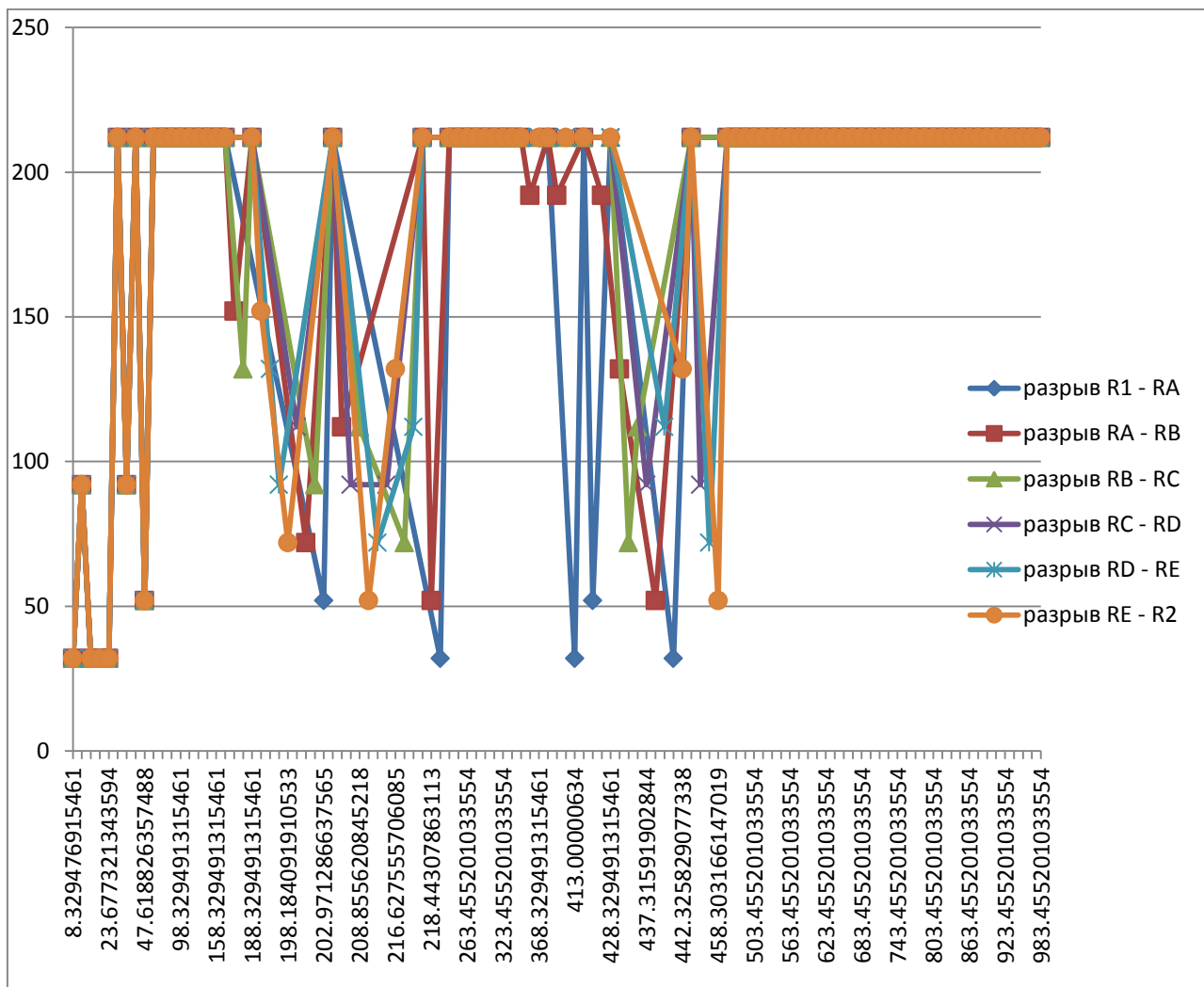


Рисунок В.1 – График прохождения байтпакетов на маршрутизаторе R1 при обрыве линии в период времени 183 с - 413 с

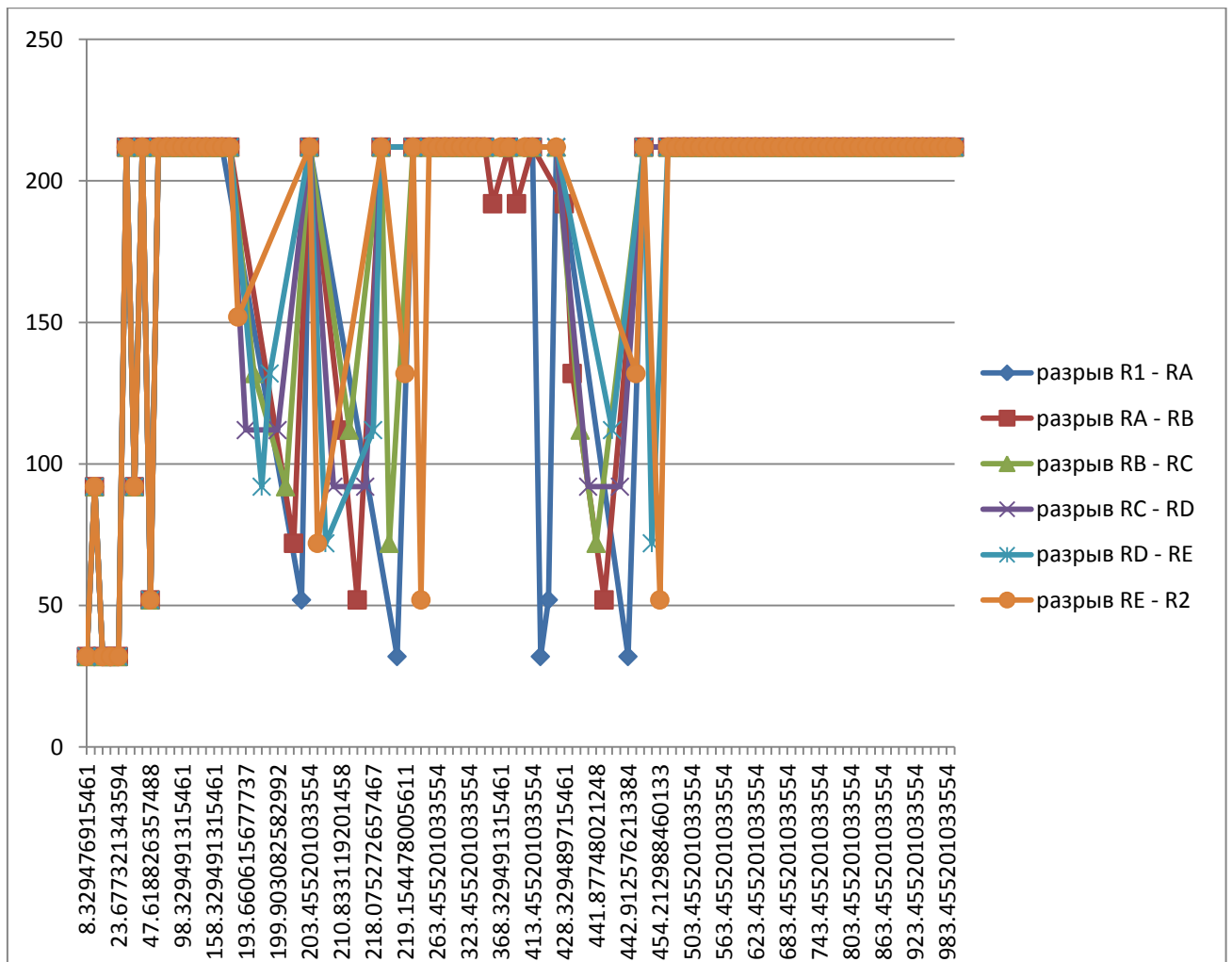


Рисунок В.2 – График прохождения байтпакетов на маршрутизаторе R1 при обрыве линии в период времени 187 с - 417 с

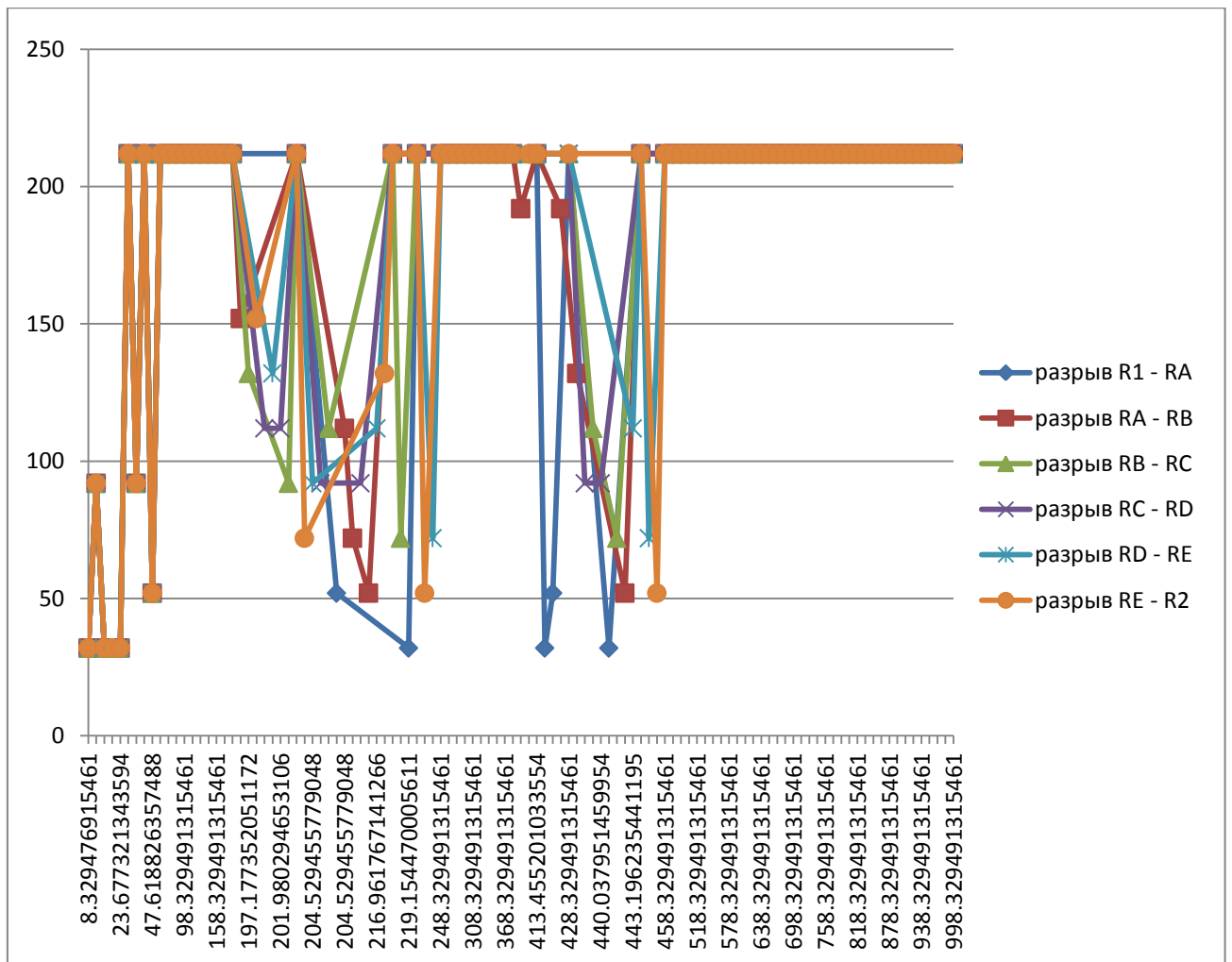


Рисунок В.3 – График прохождения байтпакетов на маршрутизаторе R1 при обрыве линии в период времени 191 с - 421 с

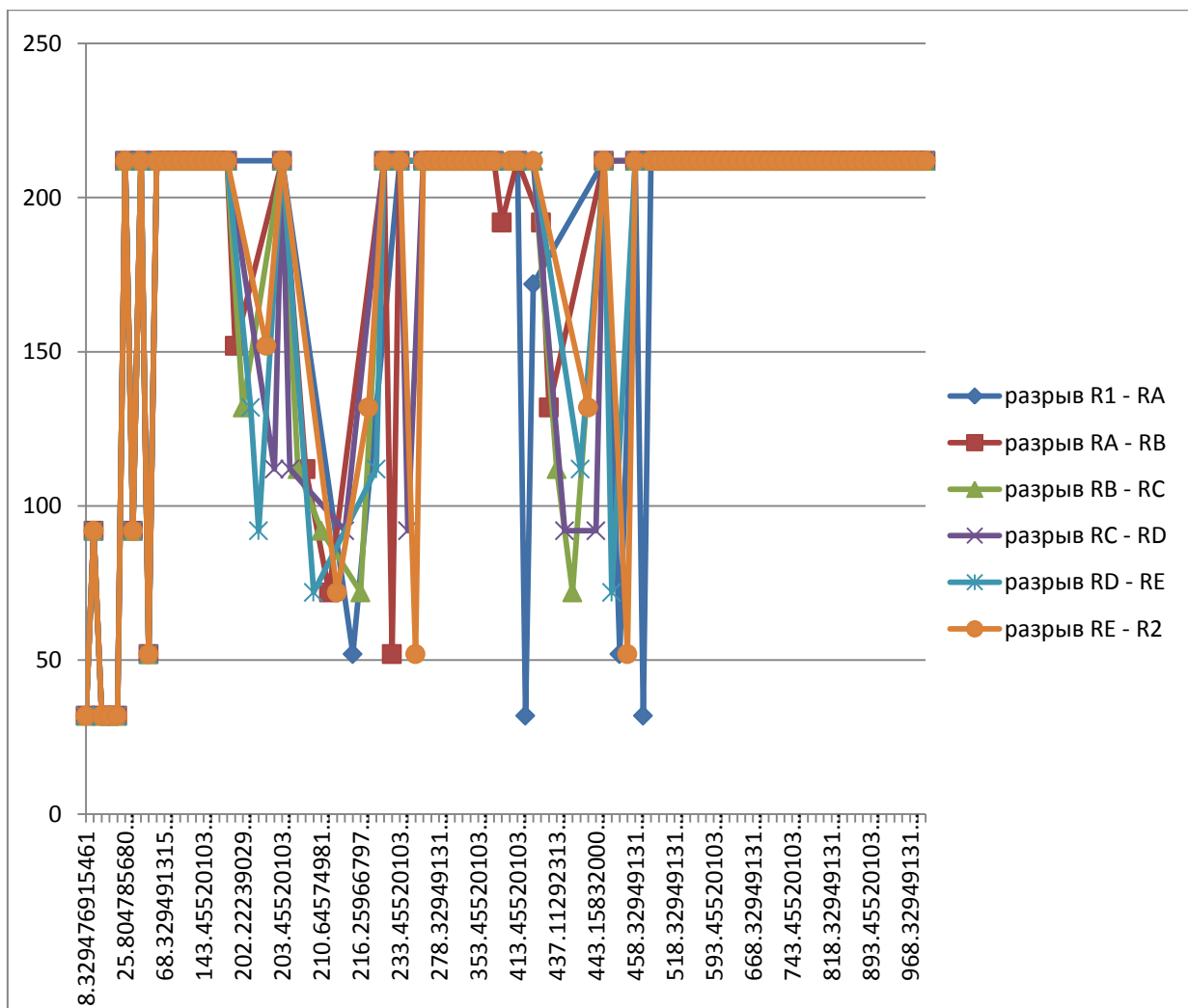


Рисунок В.4 – График прохождения байтпакетов на маршрутизаторе R1 при обрыве линии в период времени 195 с - 425 с

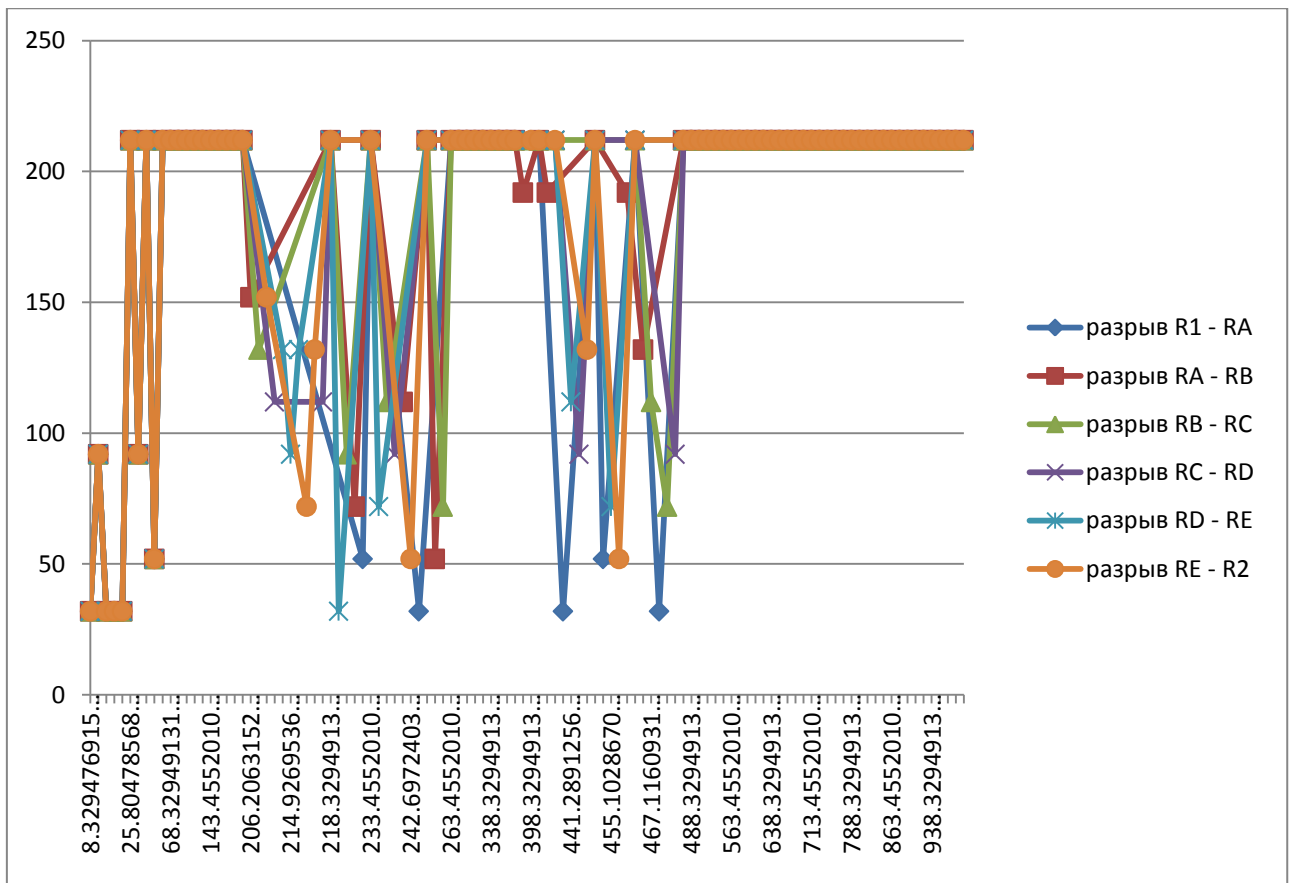


Рисунок Б.5 – График прохождения байтпакетов на маршрутизаторе R1 при обрыве линии в период времени 204 с - 434 с

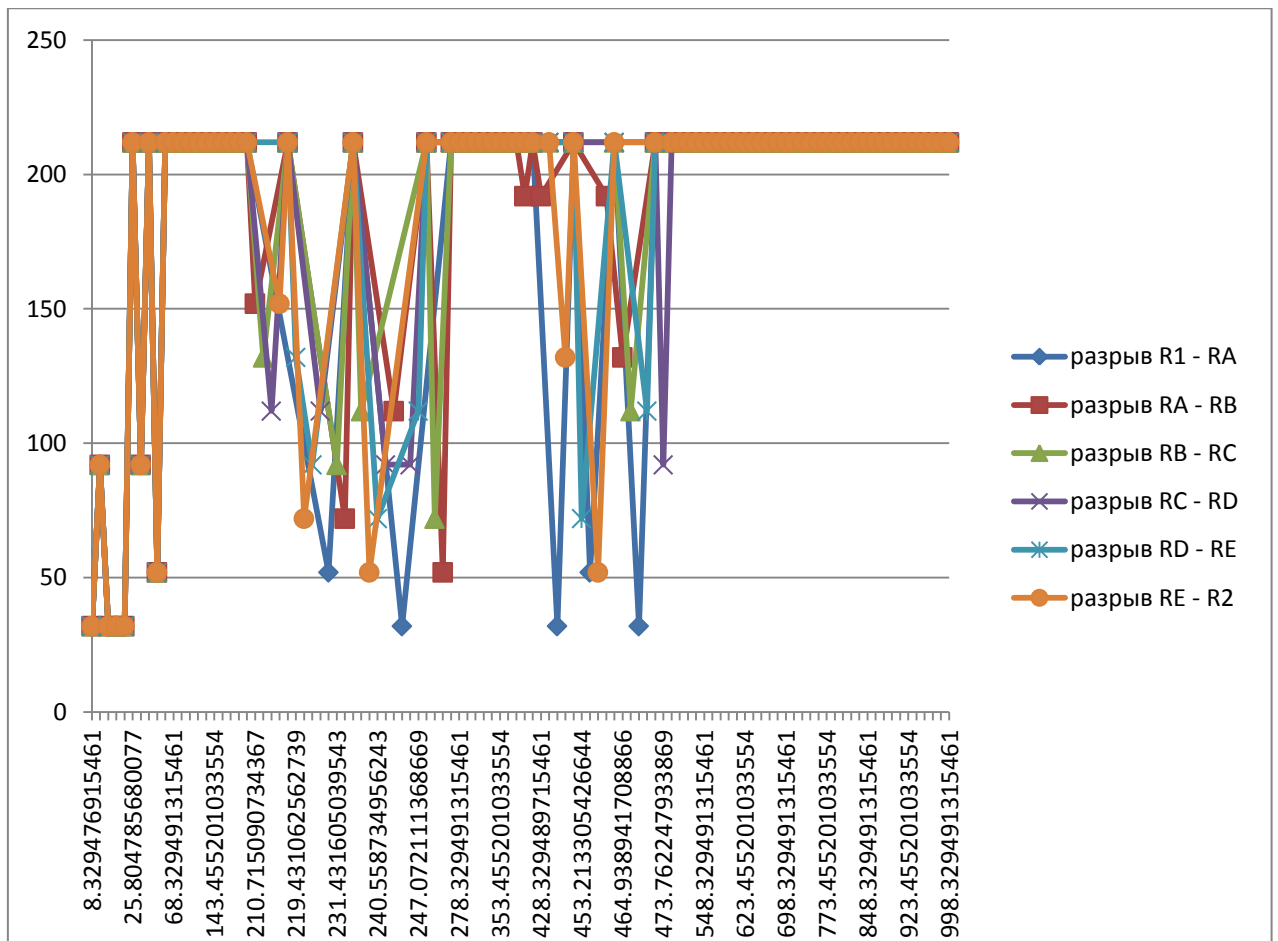


Рисунок В.6 – График прохождения байтпакетов на маршрутизаторе R1 при обрыве линии в период времени 208 с - 438 с



Рисунок В.7 – График прохождения байтпакетов на маршрутизаторе R1 при обрыве линии в период времени 212 с - 442 с

ПРИЛОЖЕНИЕ Г

Графики результатов экспериментов при отключении роутера (п. 3.2)

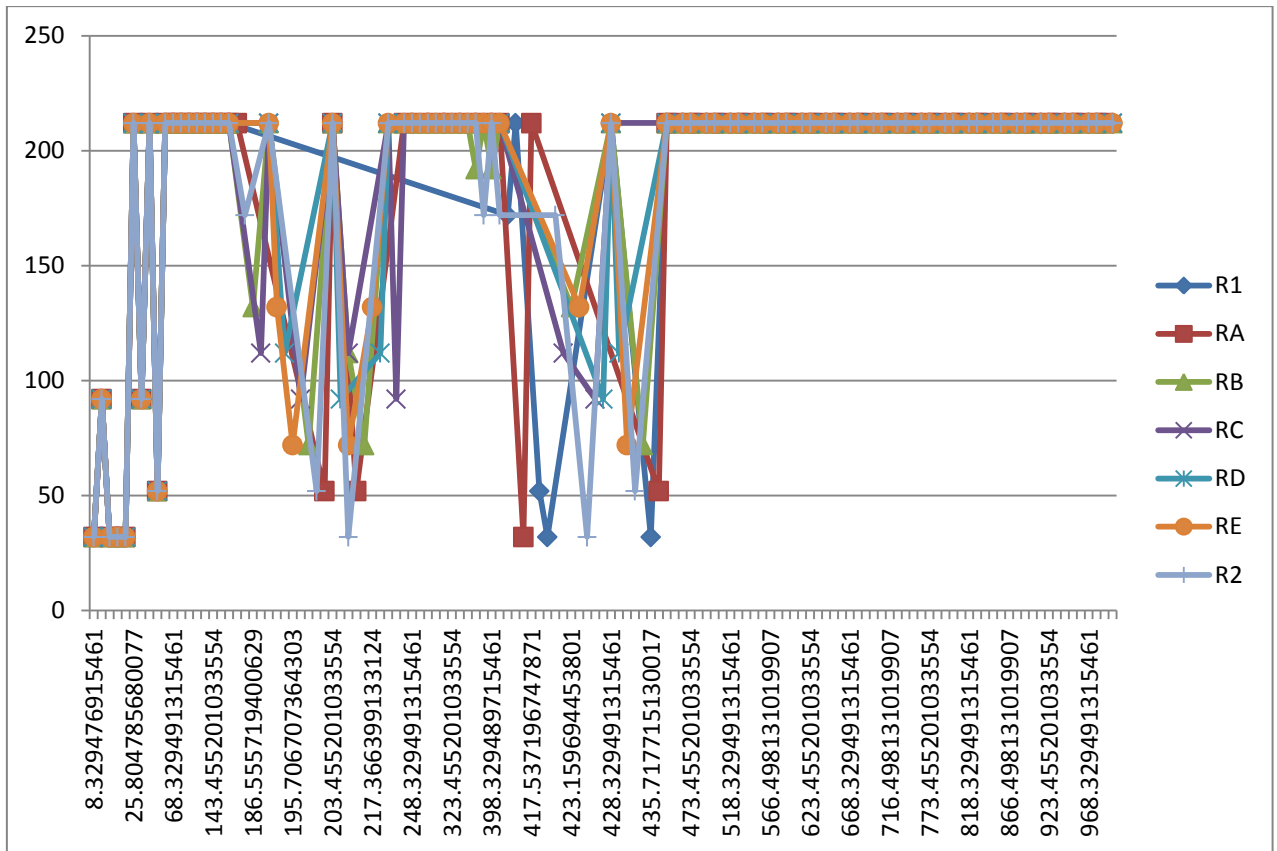


Рисунок Г.1 – График прохождения байтпакетов на маршрутизаторе R1 при поочередном отключении роутеров в период времени 183 с - 413 с

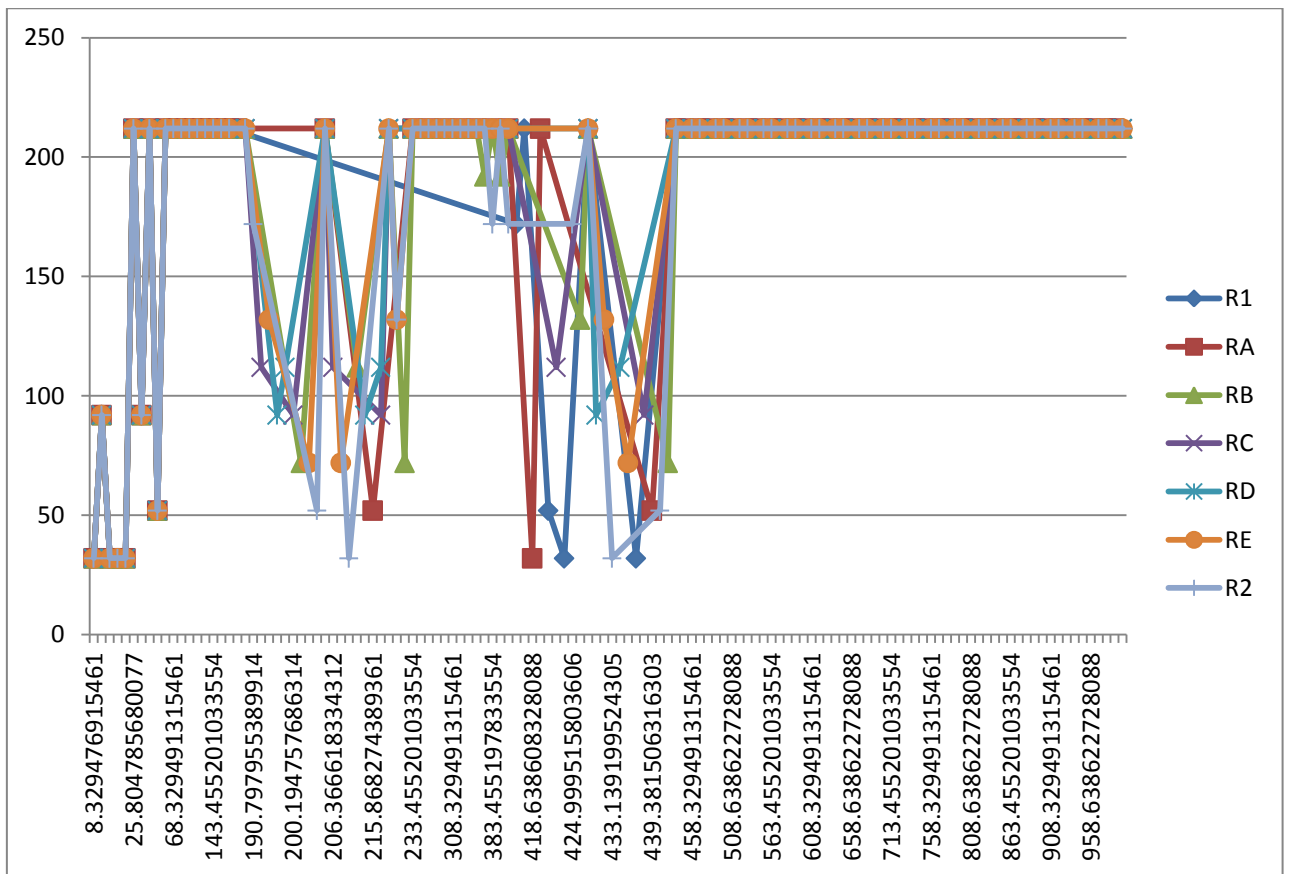


Рисунок Г.2 – График прохождения байтпакетов на маршрутизаторе R1 при поочередном отключении роутеров в период времени 187 с - 417 с

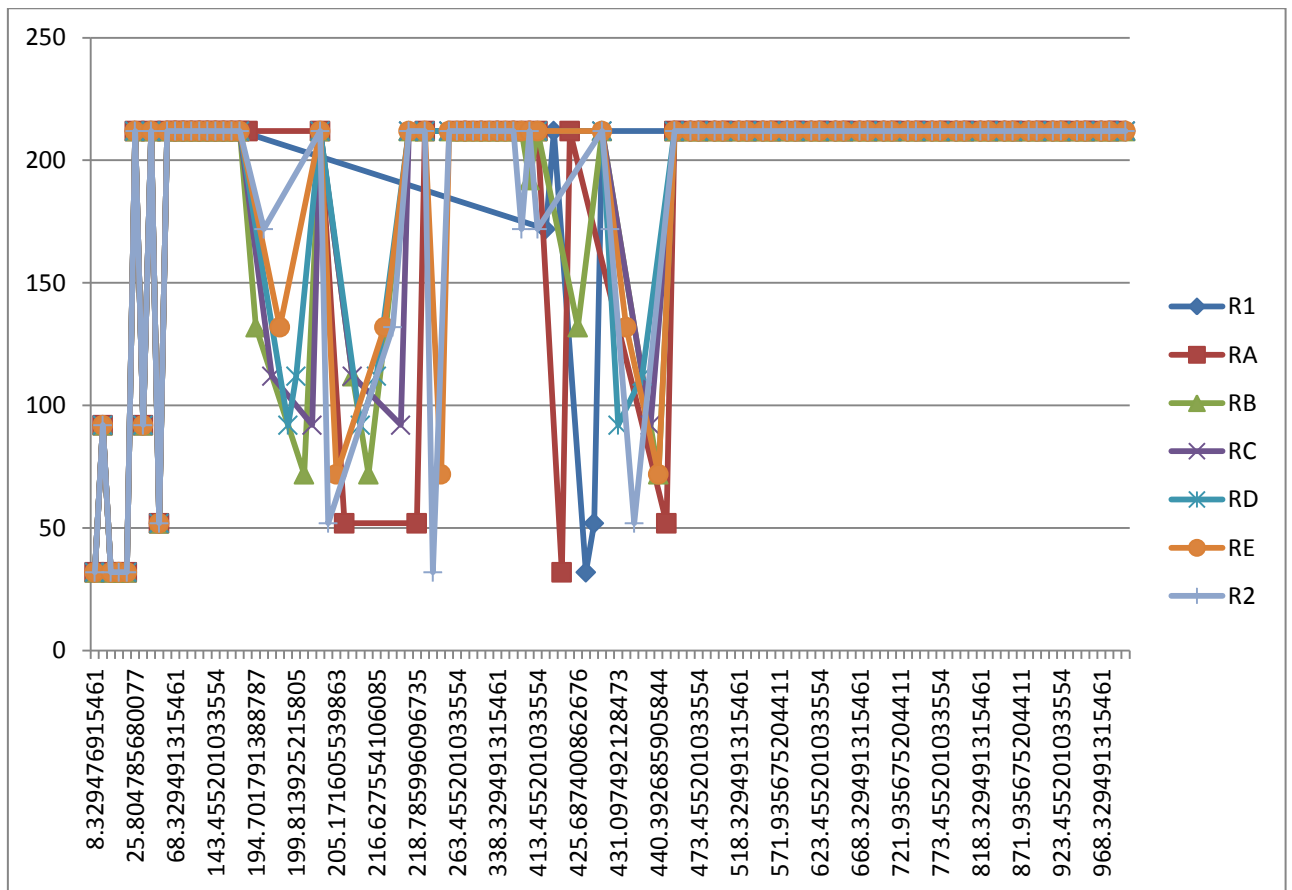


Рисунок Г.3 – График прохождения байтпакетов на маршрутизаторе R1 при поочередном отключении роутеров в период времени 191 с - 421 с

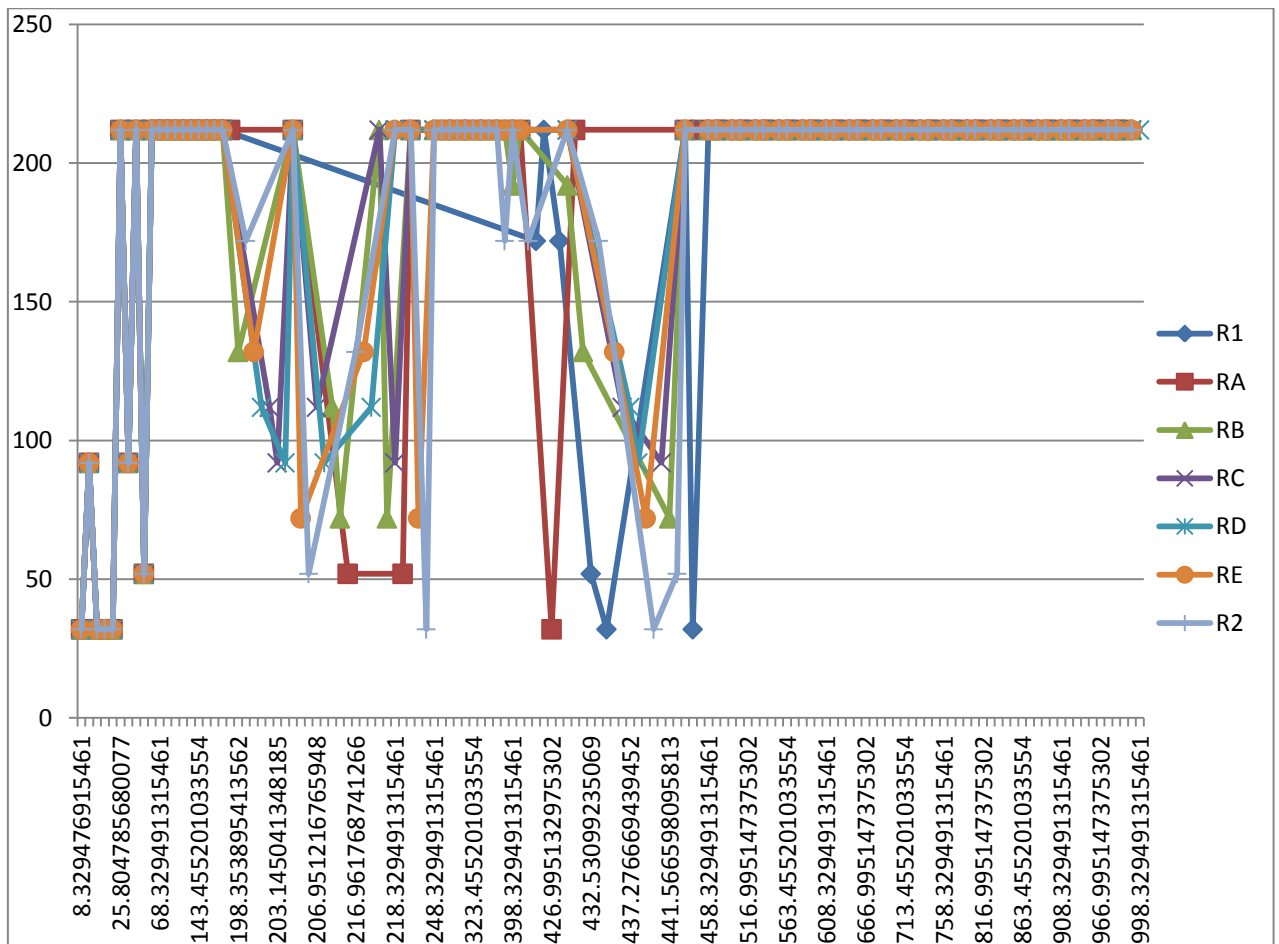


Рисунок Г.4 – График прохождения байтпакетов на маршрутизаторе R1 при поочередном отключении роутеров в период времени 195 с - 425 с

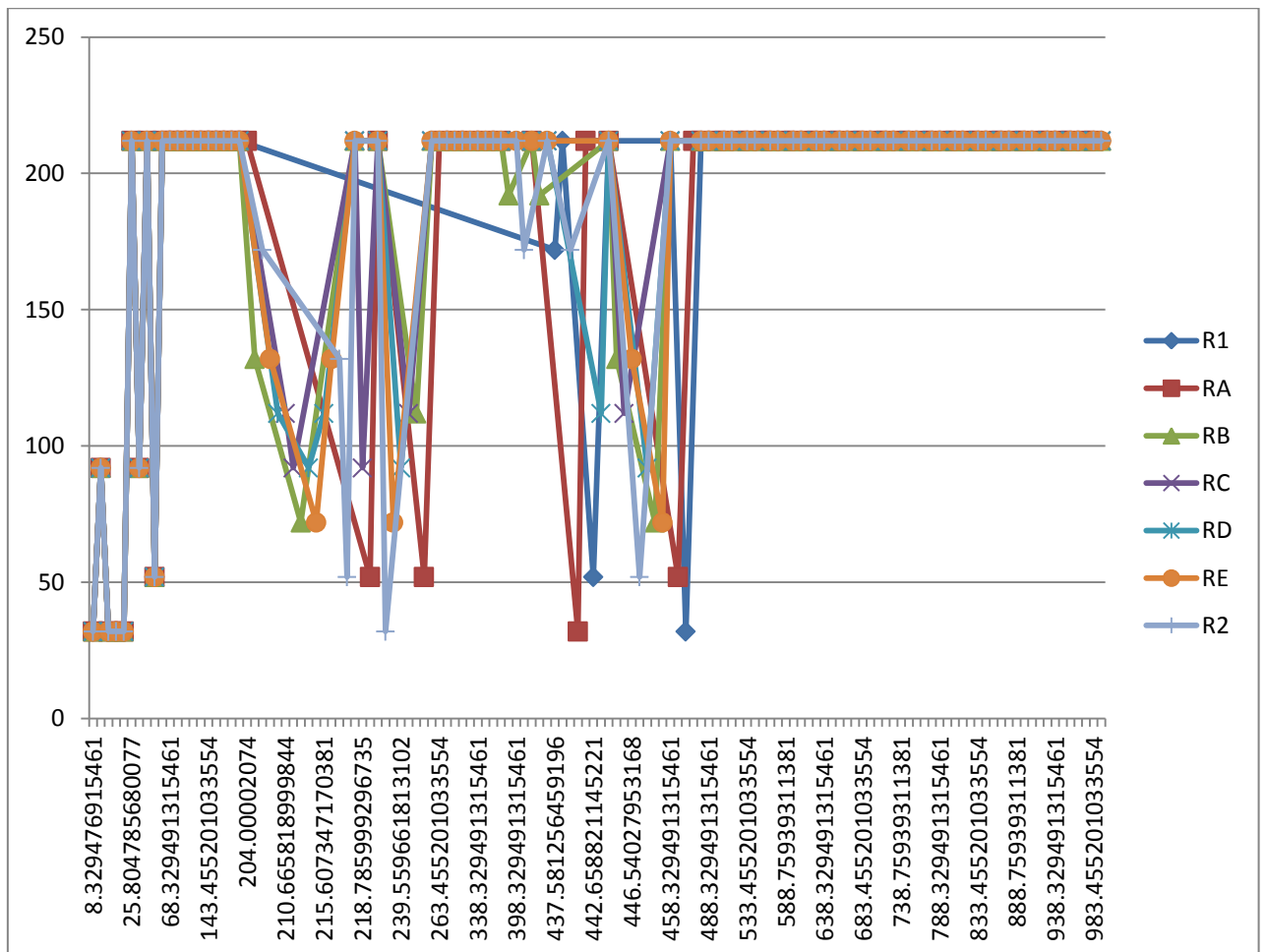


Рисунок Г.5 – График прохождения байтпакетов на маршрутизаторе R1 при поочередном отключении роутеров в период времени 204 с - 434 с

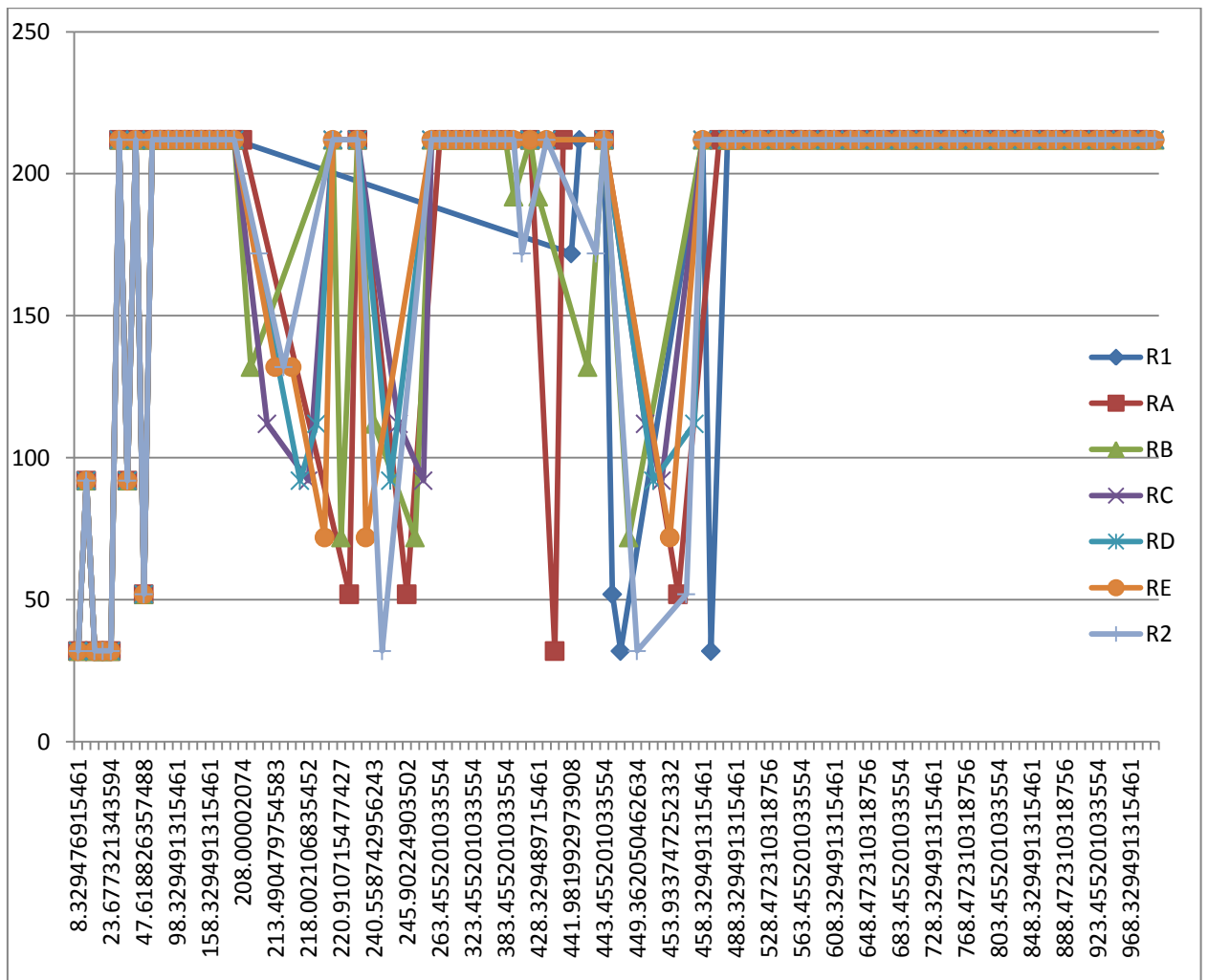


Рисунок Г.6 – График прохождения байтпакетов на маршрутизаторе R1 при поочередном отключении роутеров в период времени 208 с - 438 с

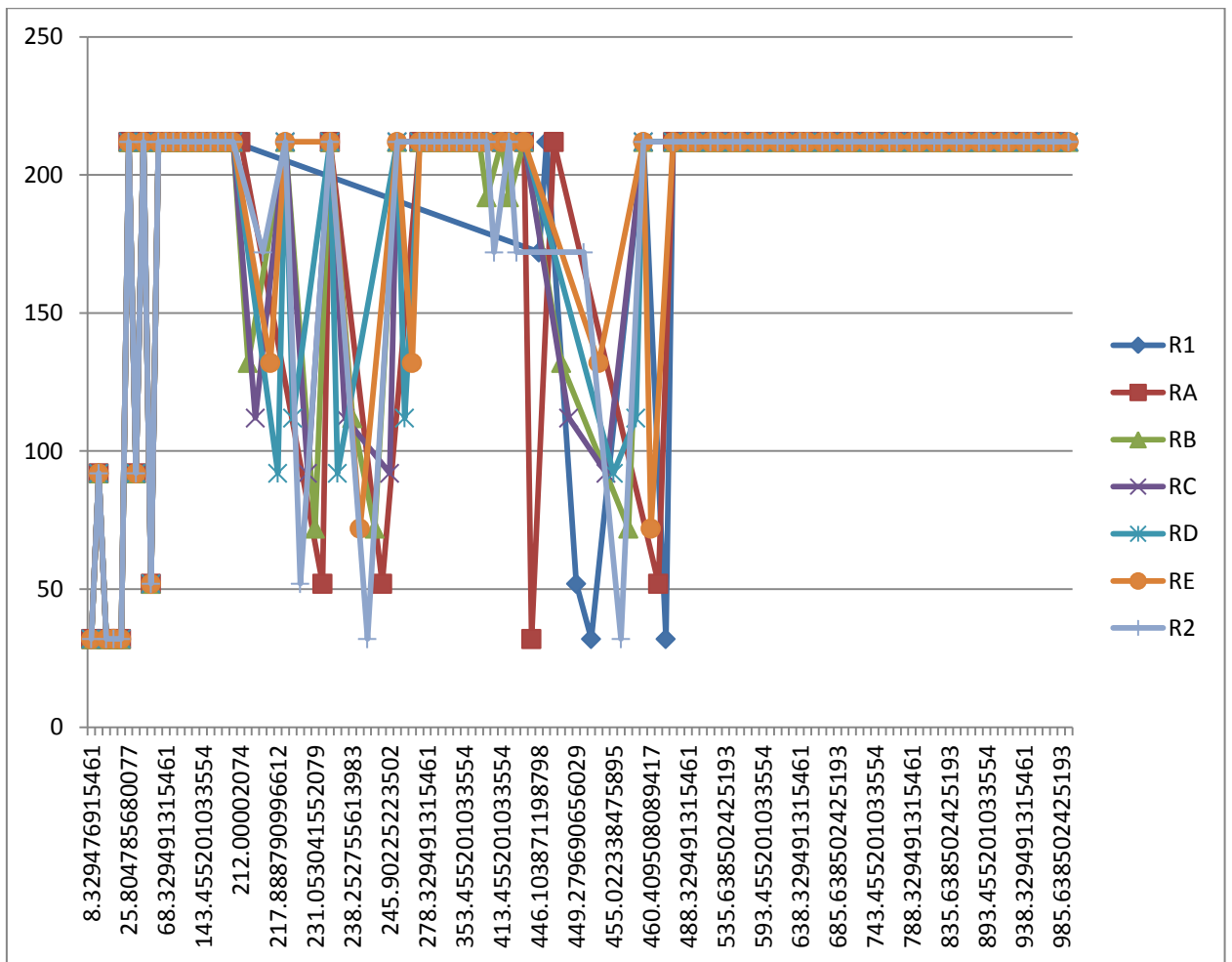


Рисунок Г.7 – График прохождения байтпакетов на маршрутизаторе R1 при поочередном отключении роутеров в период времени 212 с - 442 с